Fig. 1

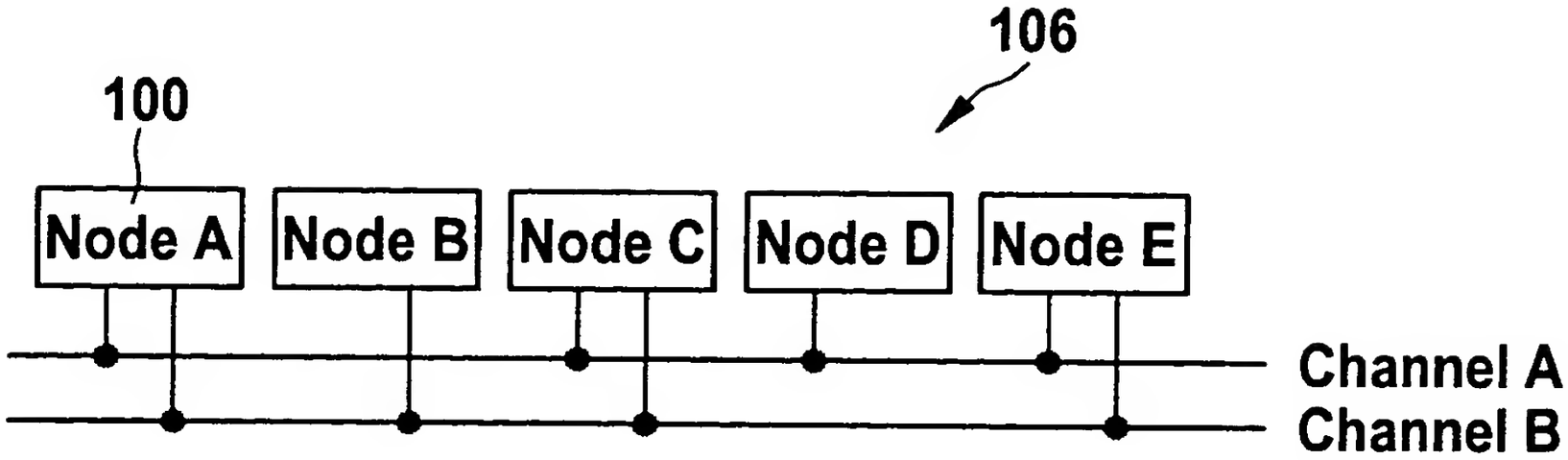


Fig. 2

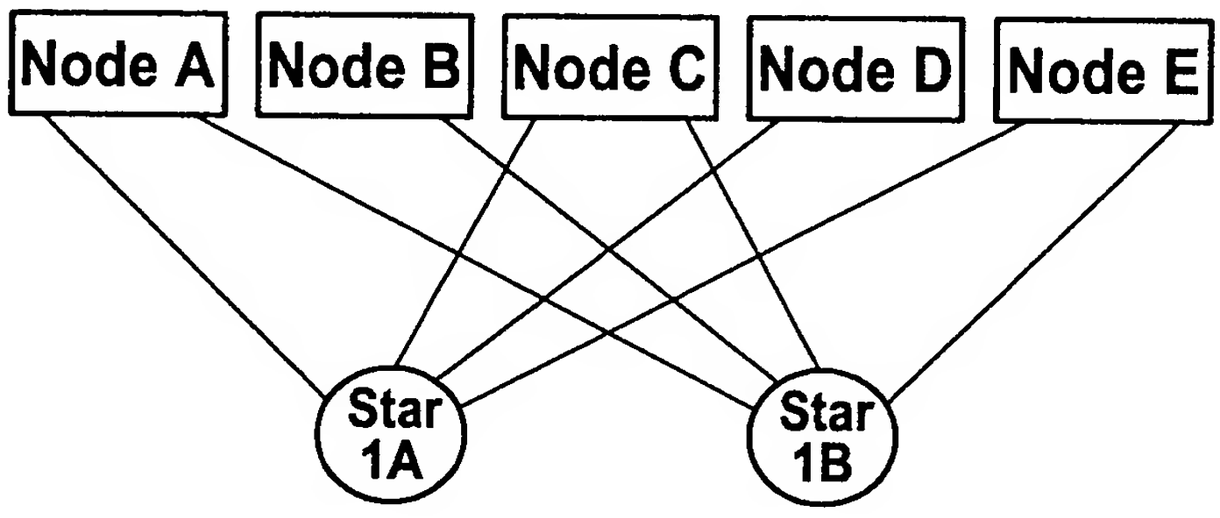


Fig. 3

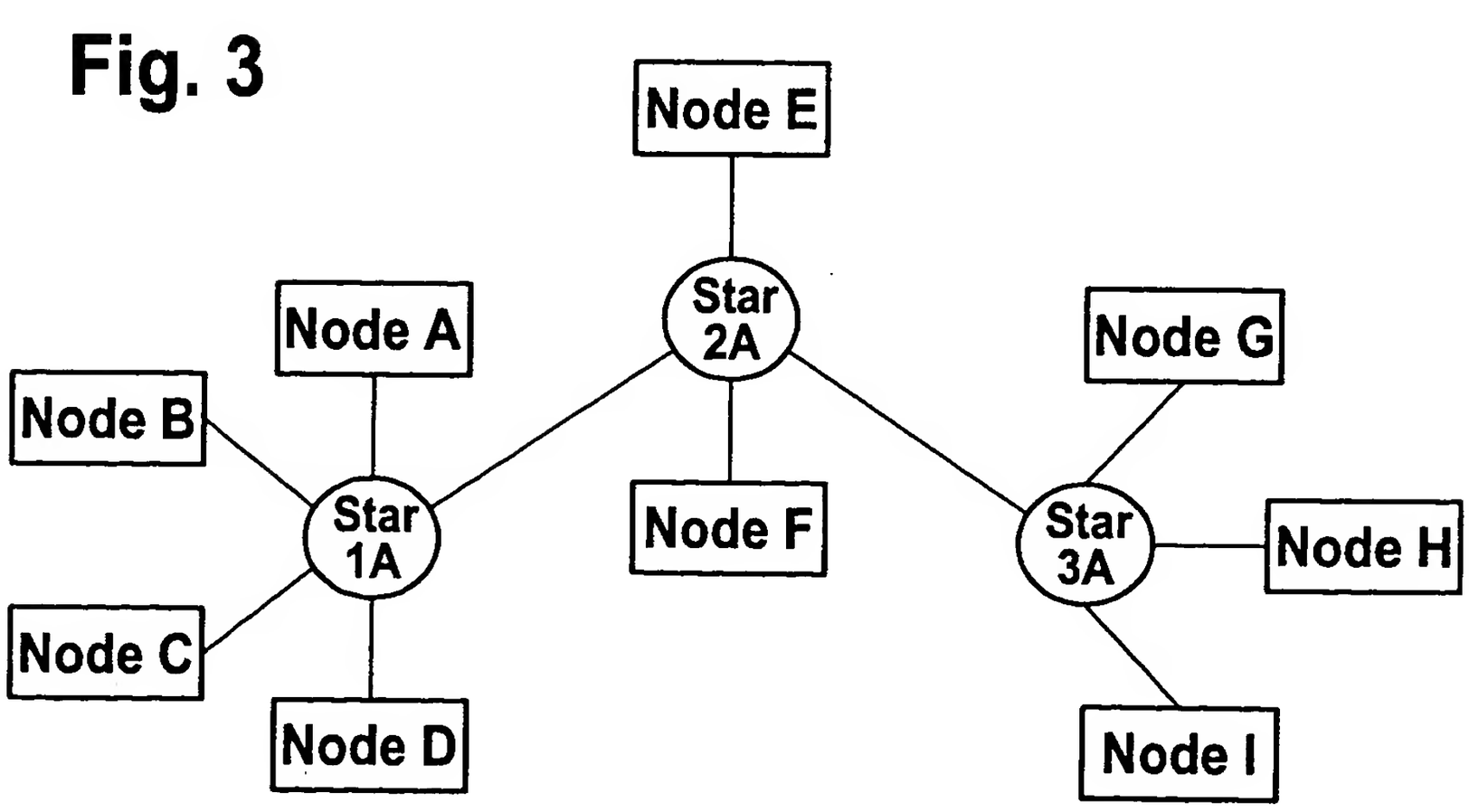


Fig. 4

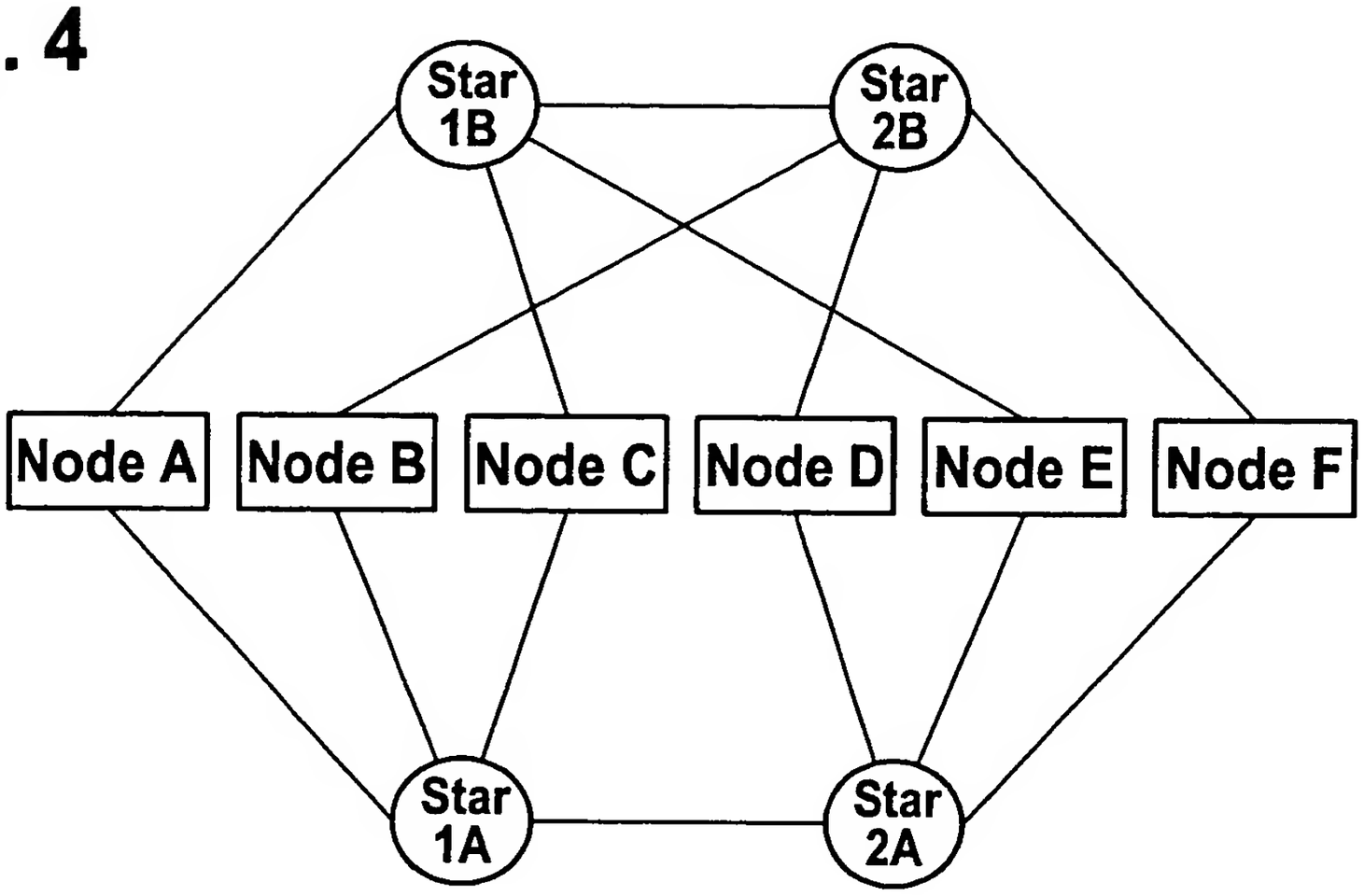

Star 1B
Star 2B
Node A
Node B
Node C
Node D
Node E
Node F
Star 1A
Star 2A

Fig. 5

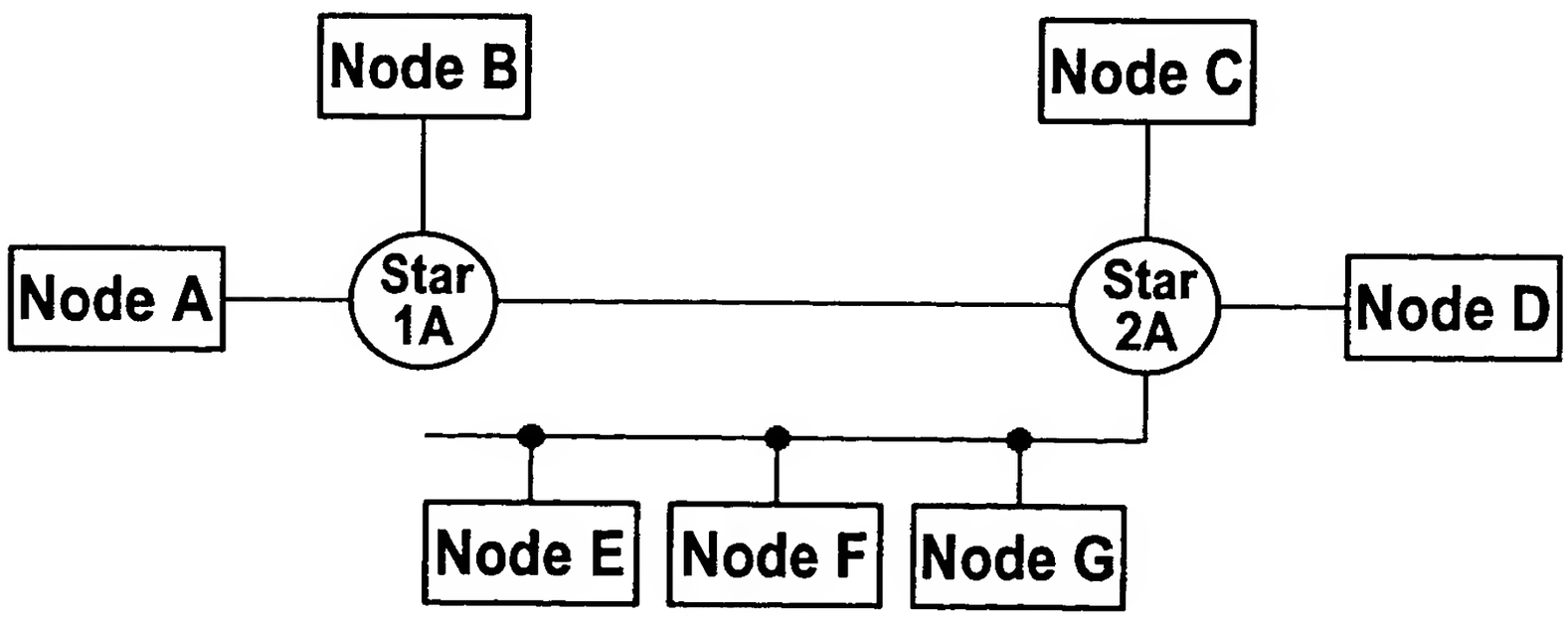

Node B
Node C
Node A
Star 1A
Star 2A
Node D
Node E
Node F
Node G

Fig. 6

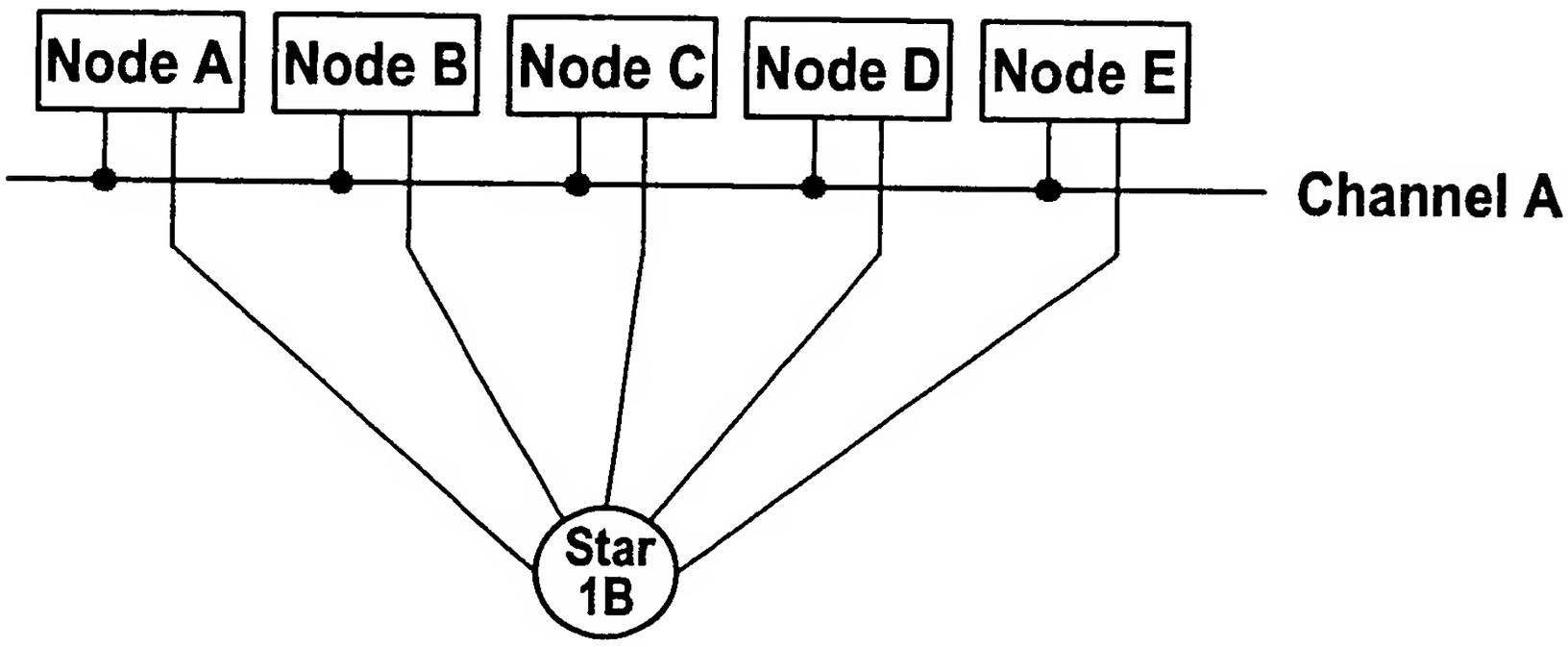

Node A
Node B
Node C
Node D
Node E
Channel A
Star
1B

Fig. 7

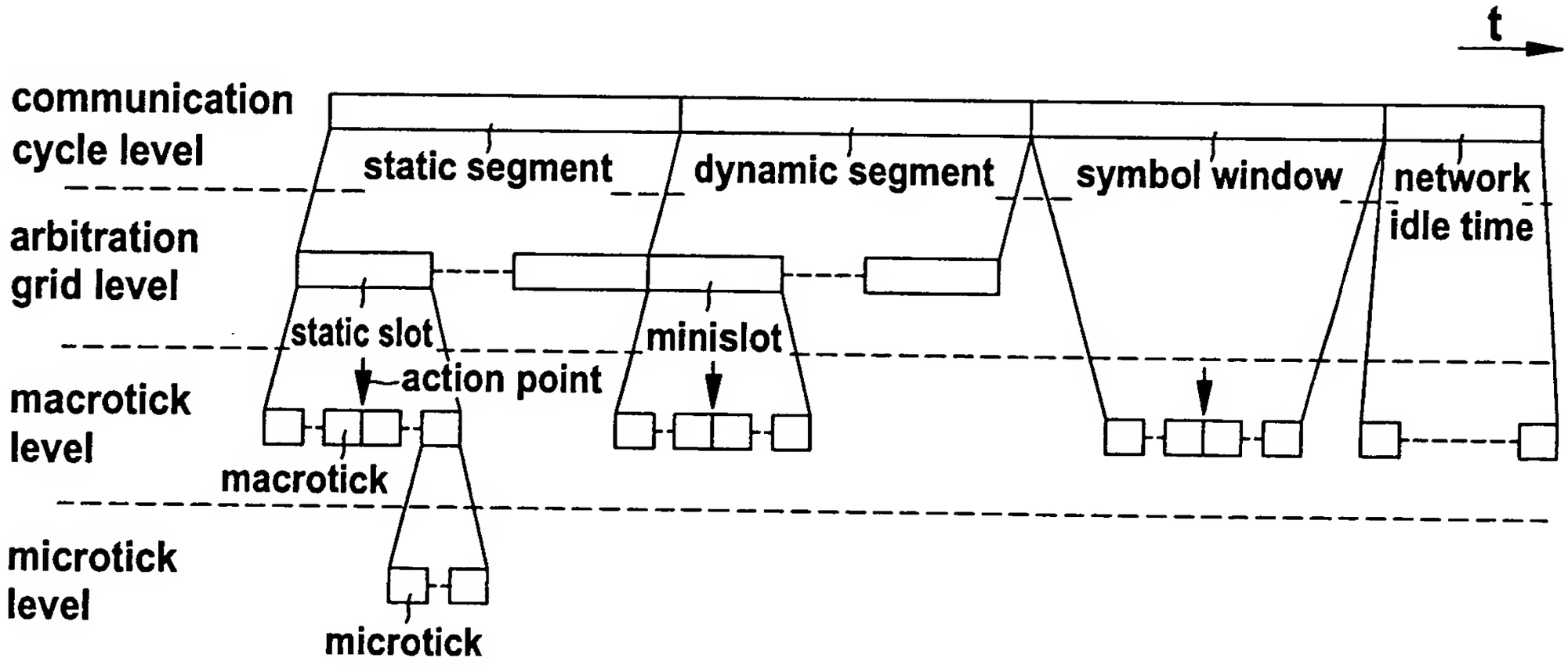

t
communication
cycle level
static segment
dynamic segment
symbol window
network
idle time
arbitration
grid level
static slot
minislot
action point
macrotick
level
macrotick
microtick
level
microtick

Fig. 8

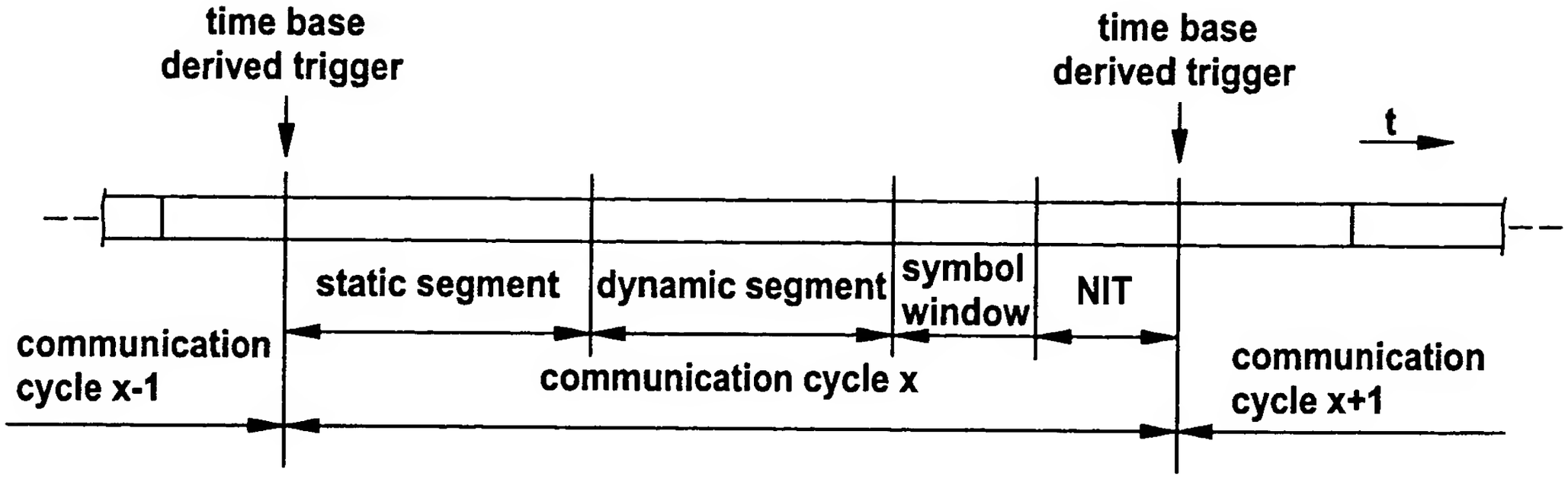


Fig. 9

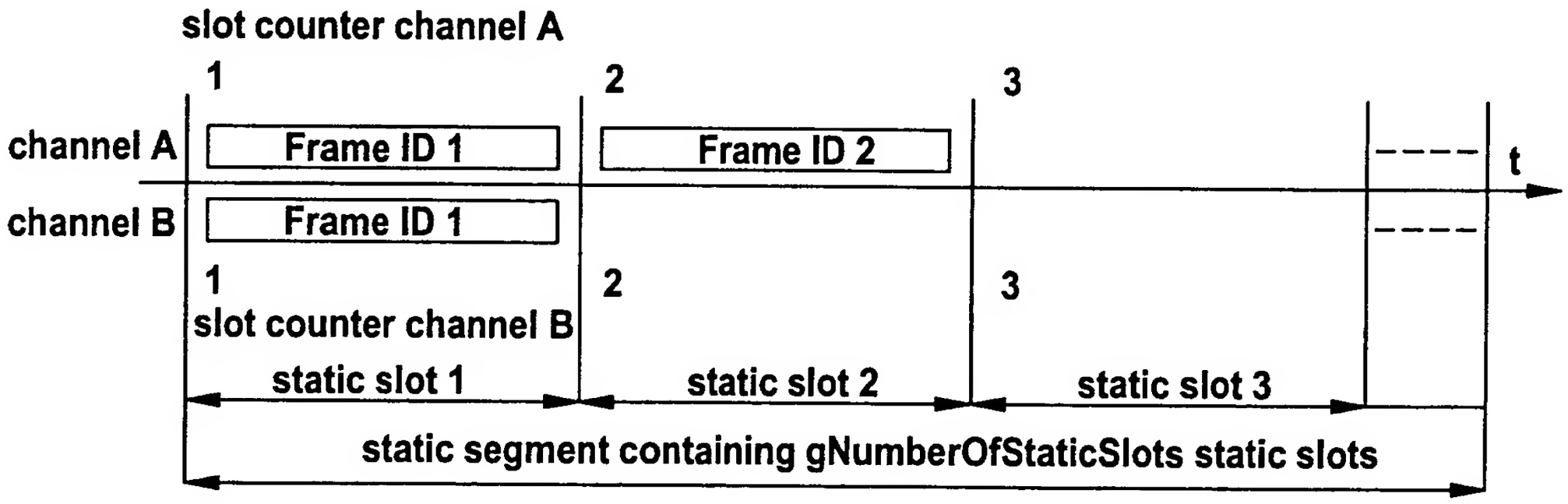
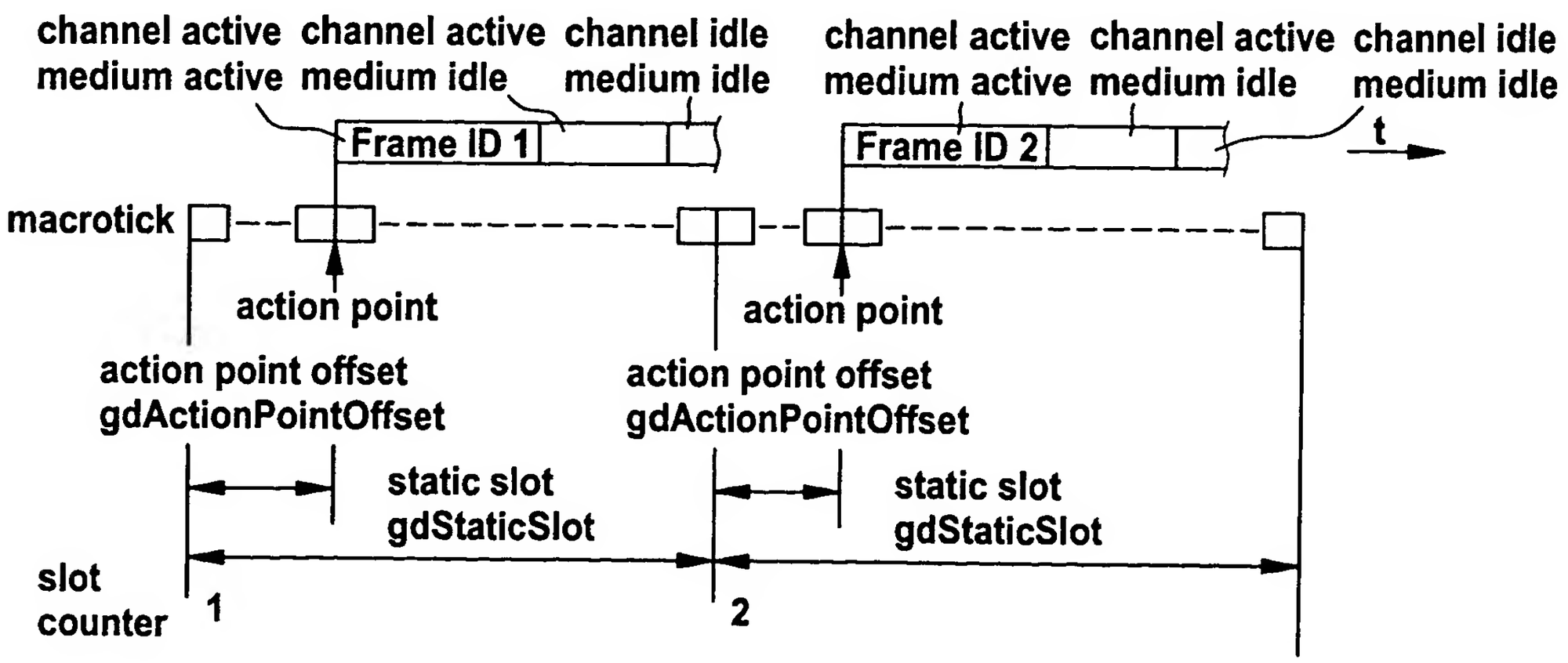


Fig. 10



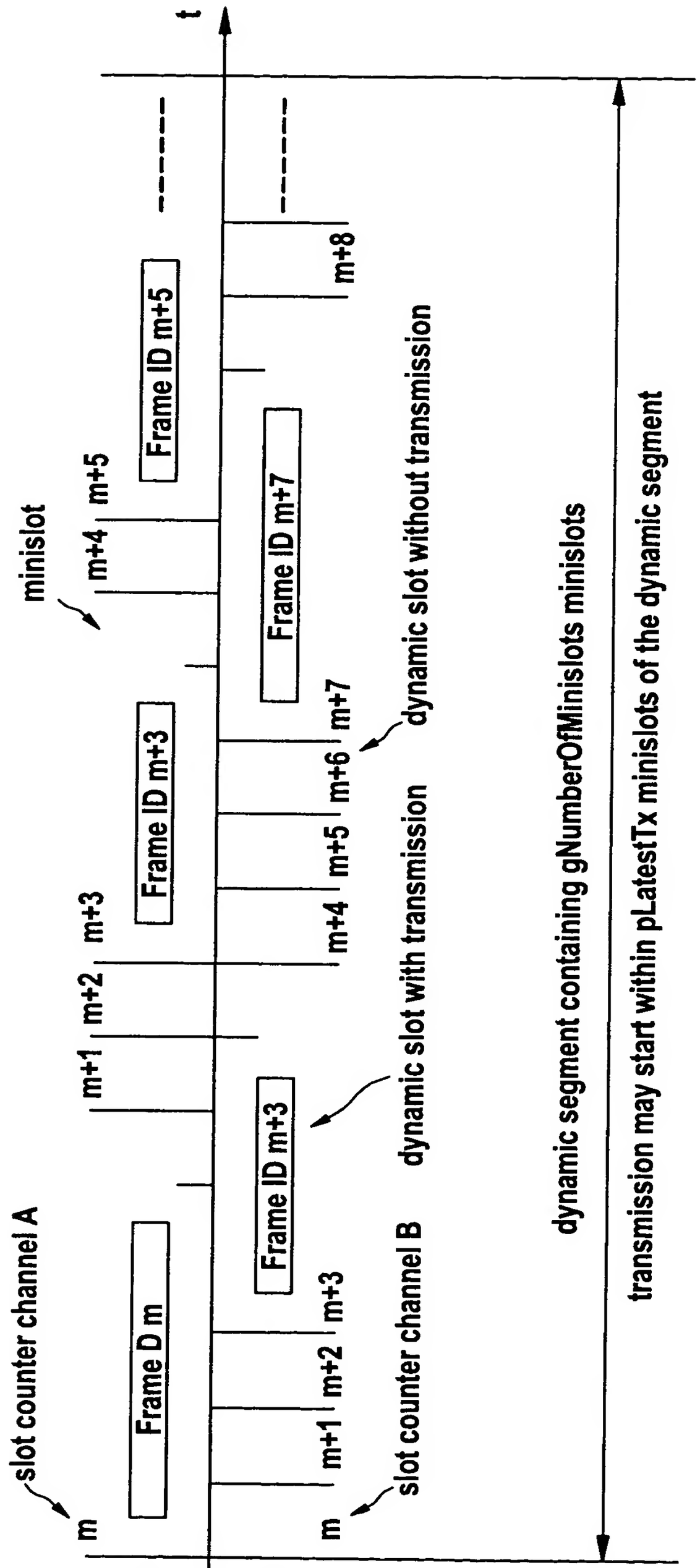
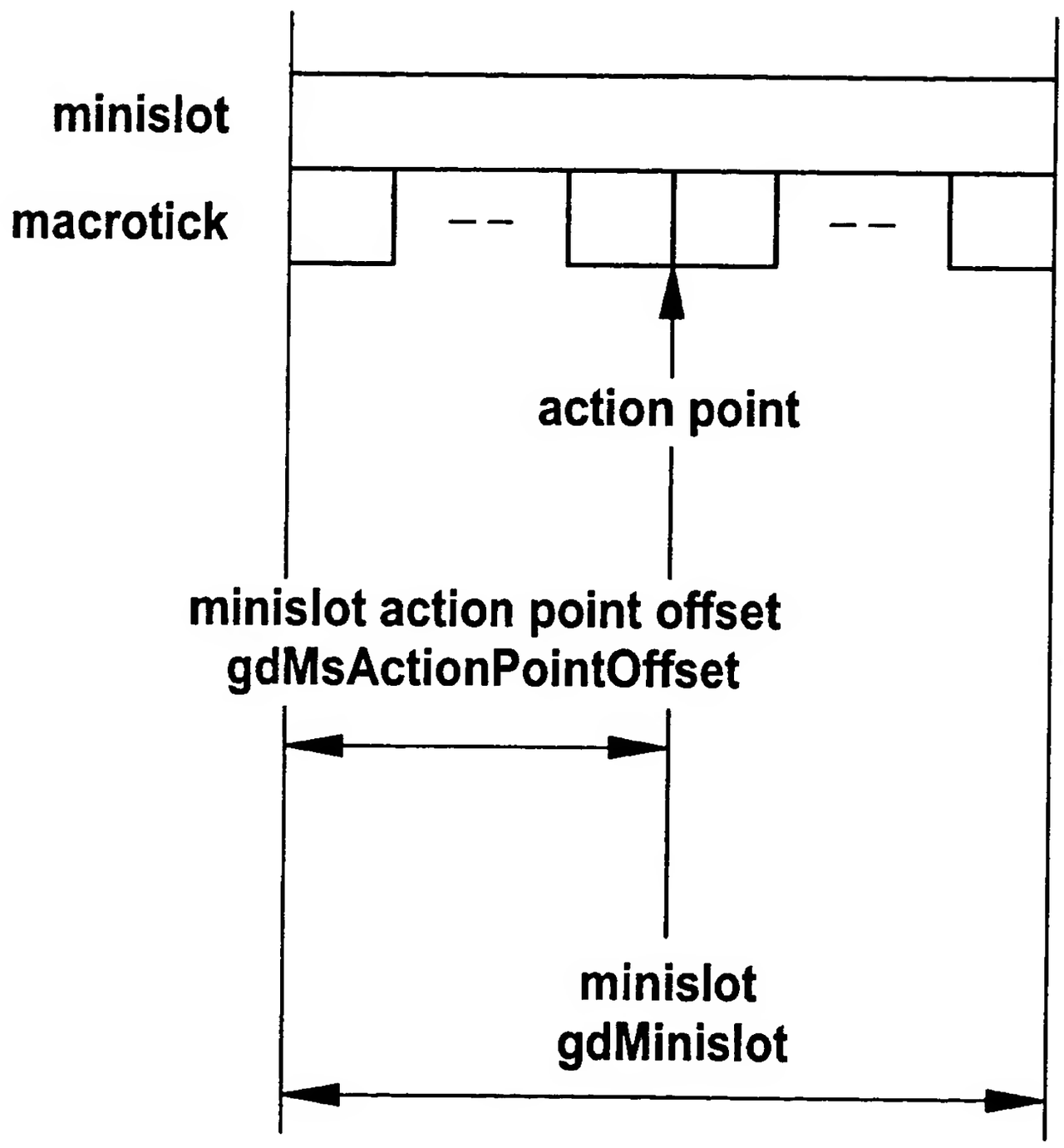

t
channel A
channel B
slot counter channel A
slot counter channel B
m
m+1
m+2
m+3
m+4
m+5
m+6
m+7
minislot
Frame ID m+3
Frame ID m+5
Frame ID m+7
dynamic slot with transmission
dynamic slot without transmission
dynamic segment containing gNumberOfMinislots minislots
transmission may start within pLatestTx minislots of the dynamic segment

Fig. 11

Fig. 12


minislot
macrotick
action point
minislot action point offset
gdMsActionPointOffset
minislot
gdMinislot

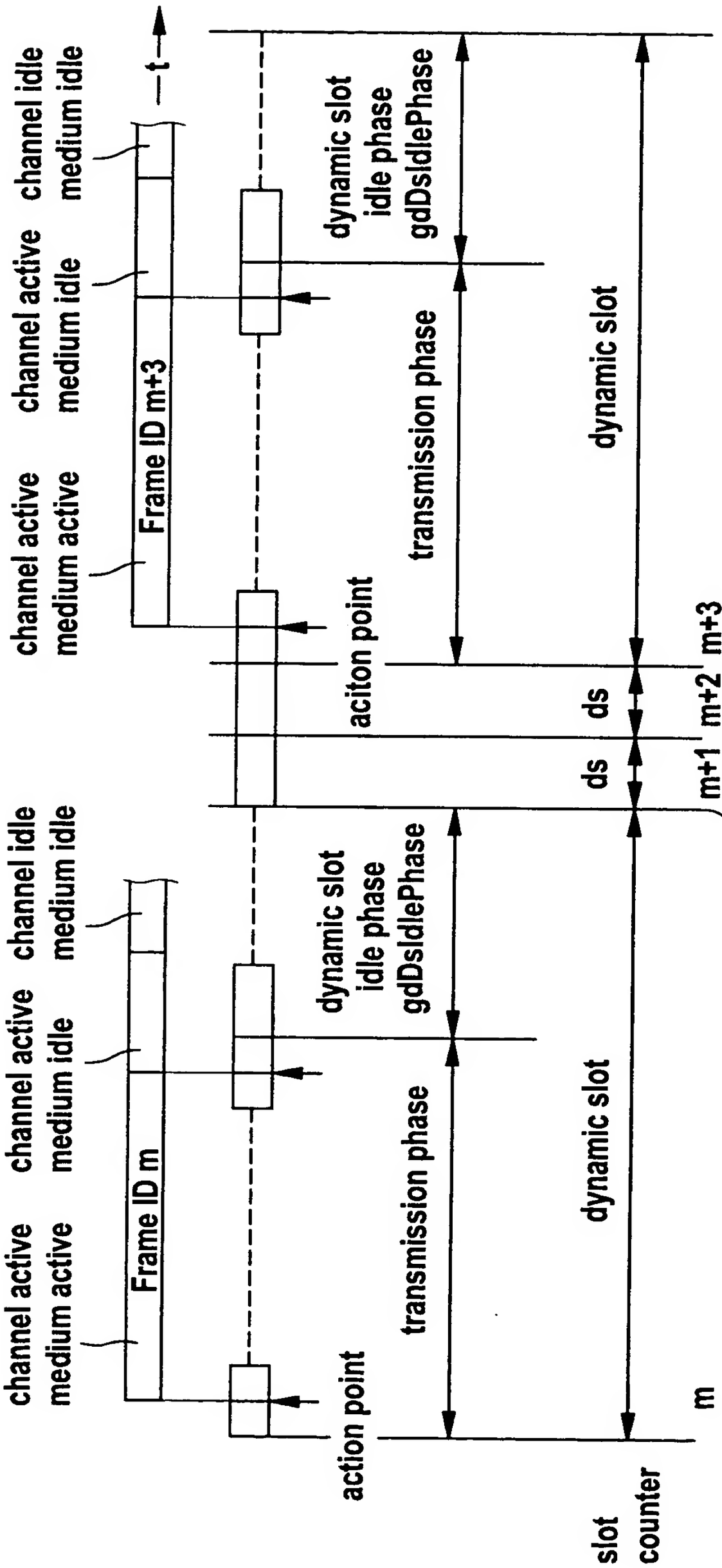

channel idle
medium idle
channel active
medium active
t
dynamic slot
idle phase
gdDsIdlePhase
Frame ID m+3
transmission phase
dynamic slot
aciton point
ds
ds
m+1
m+2
m+3
slot counter incremented at
the end of each dynamic slot
channel idle
medium idle
channel active
medium active
Frame ID m
transmission phase
dynamic slot
action point
slot
counter
m

Fig. 13

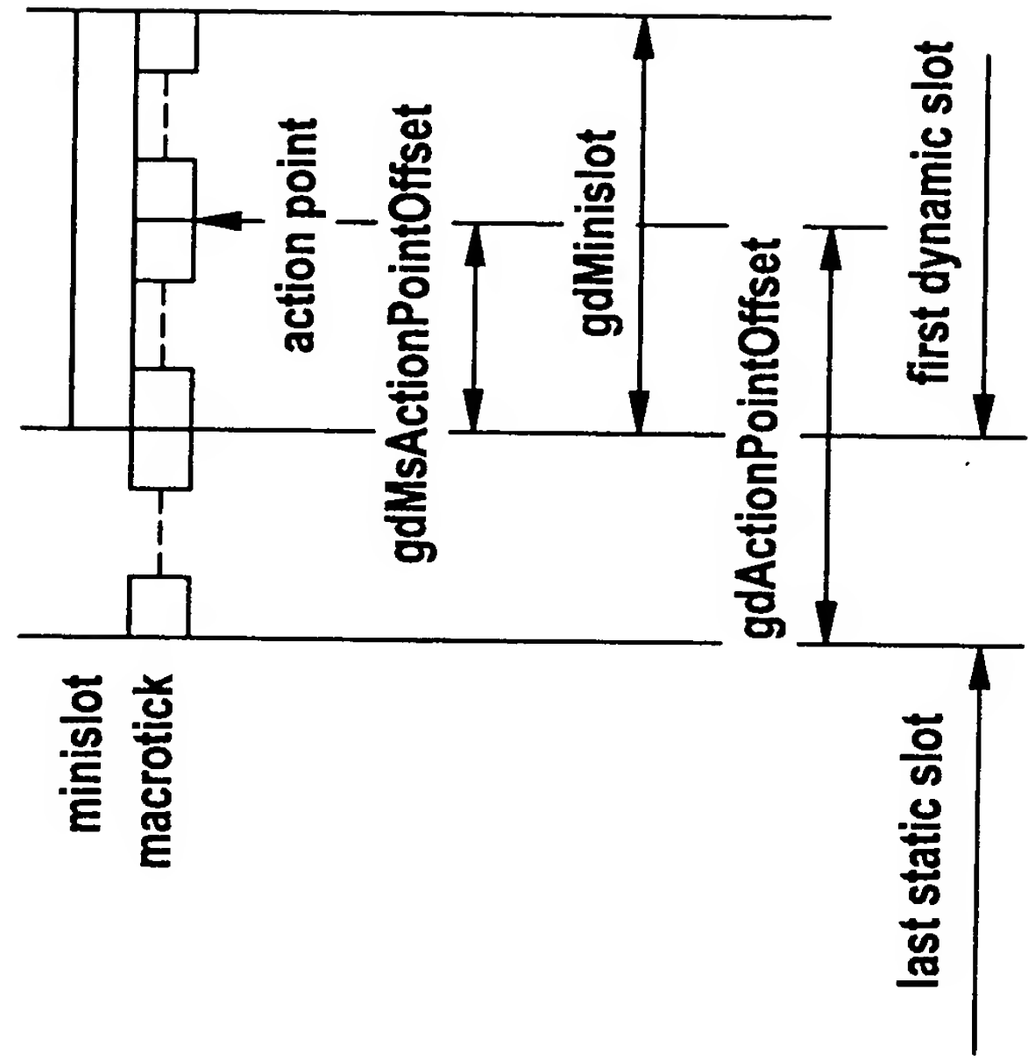

minislot
macrotick
action point
gdMsActionPointOffset
gdMinislot
gdActionPointOffset
first dynamic slot
last static slot

Fig. 14

Fig. 15

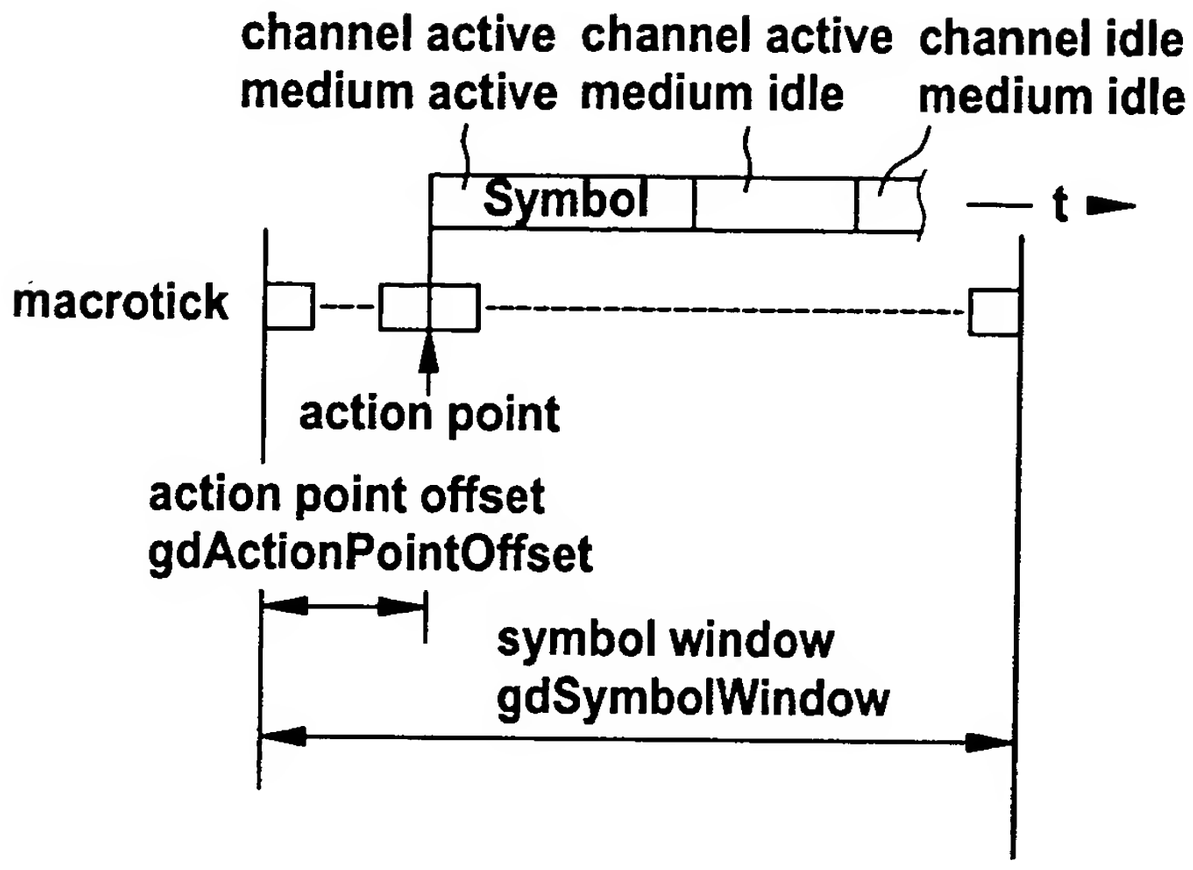

channel active
medium active
channel active
medium idle
channel idle
medium idle
Symbol
t
macrotick
action point
action point offset
gdActionPointOffset
symbol window
gdSymbolWindow

Fig. 16

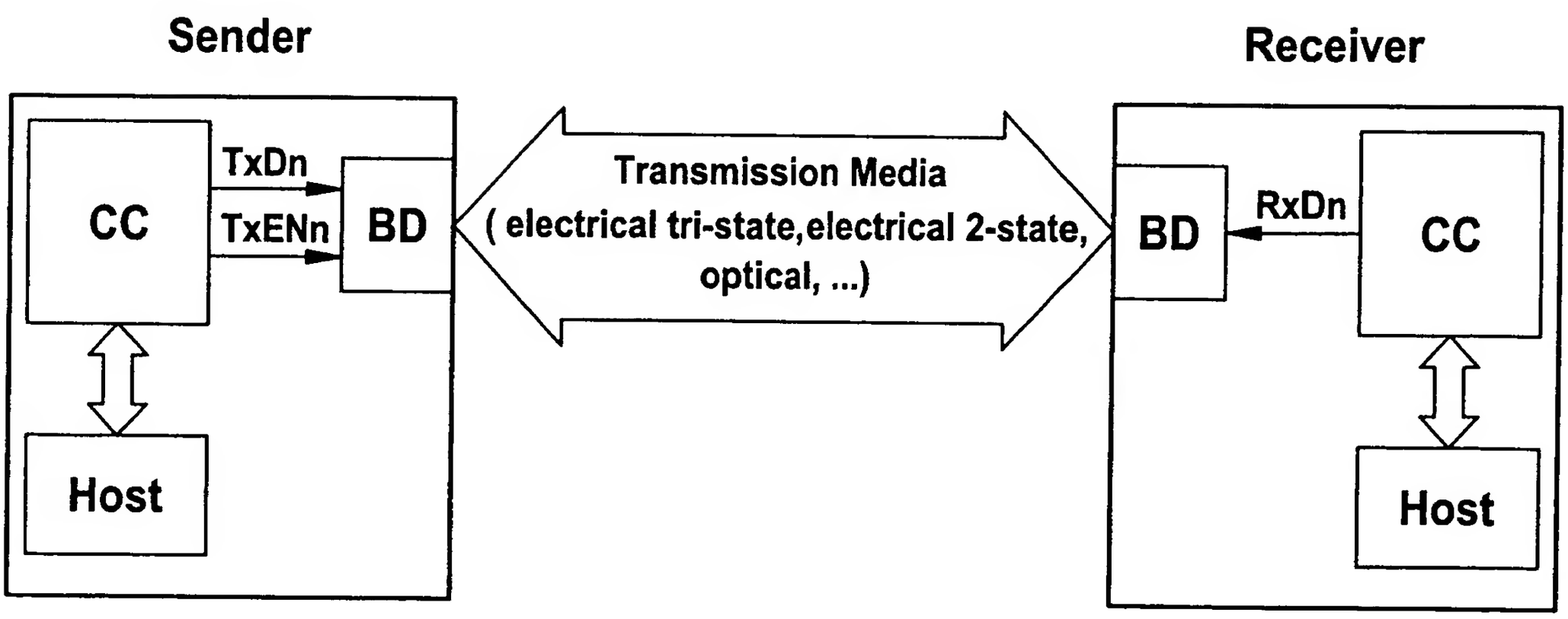

Sender
Receiver
CC
TxDn
TxENn
BD
Transmission Media
(electrical tri-state,electrical 2-state,
optical, ...)
BD
RxDn
CC
Host
Host

Fig. 17

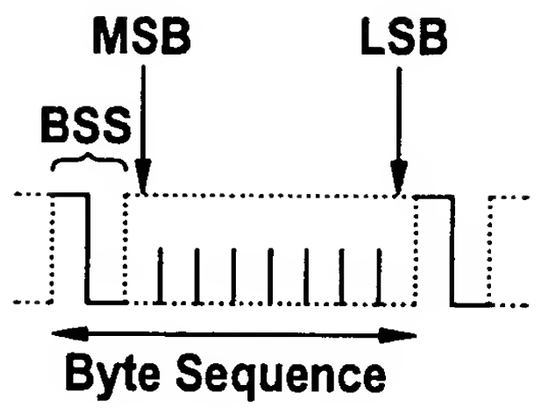


Fig. 18

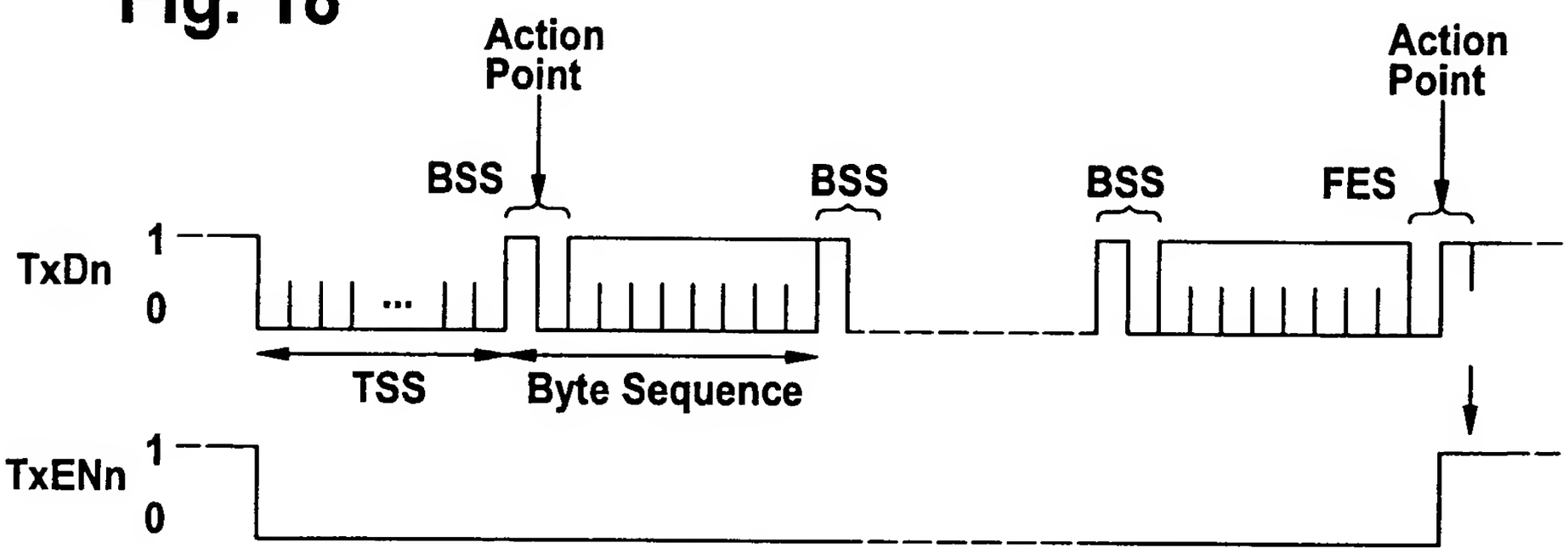


Fig. 19

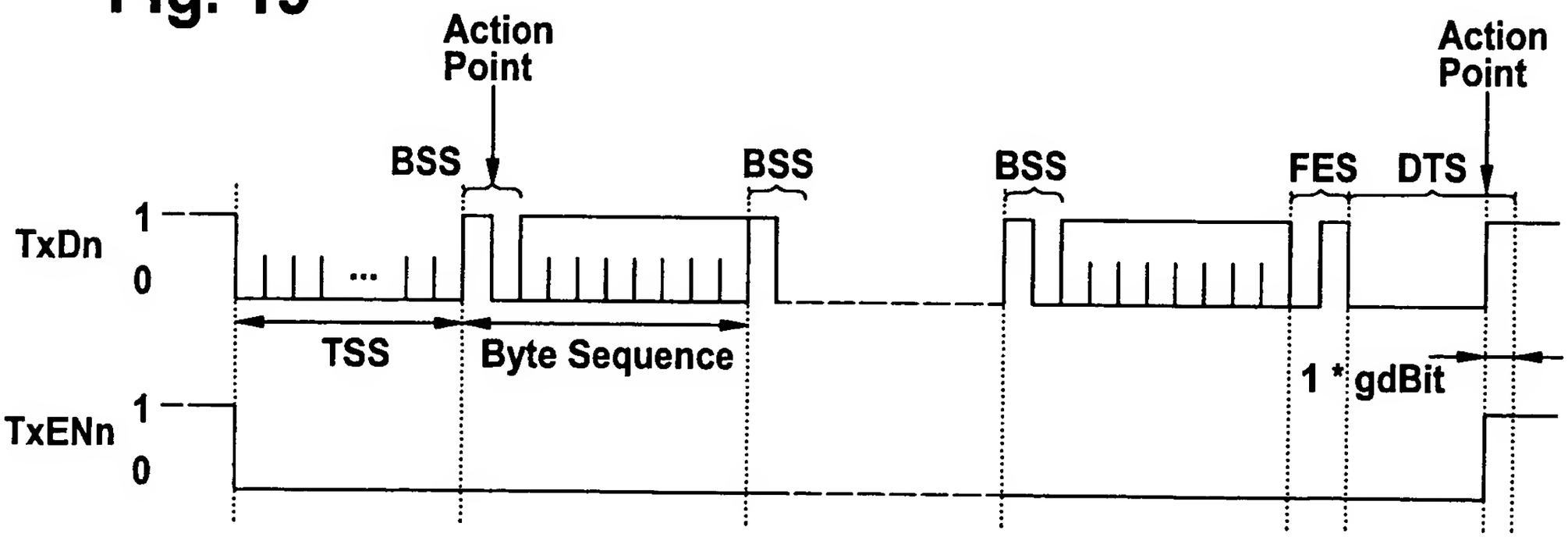


Fig. 20

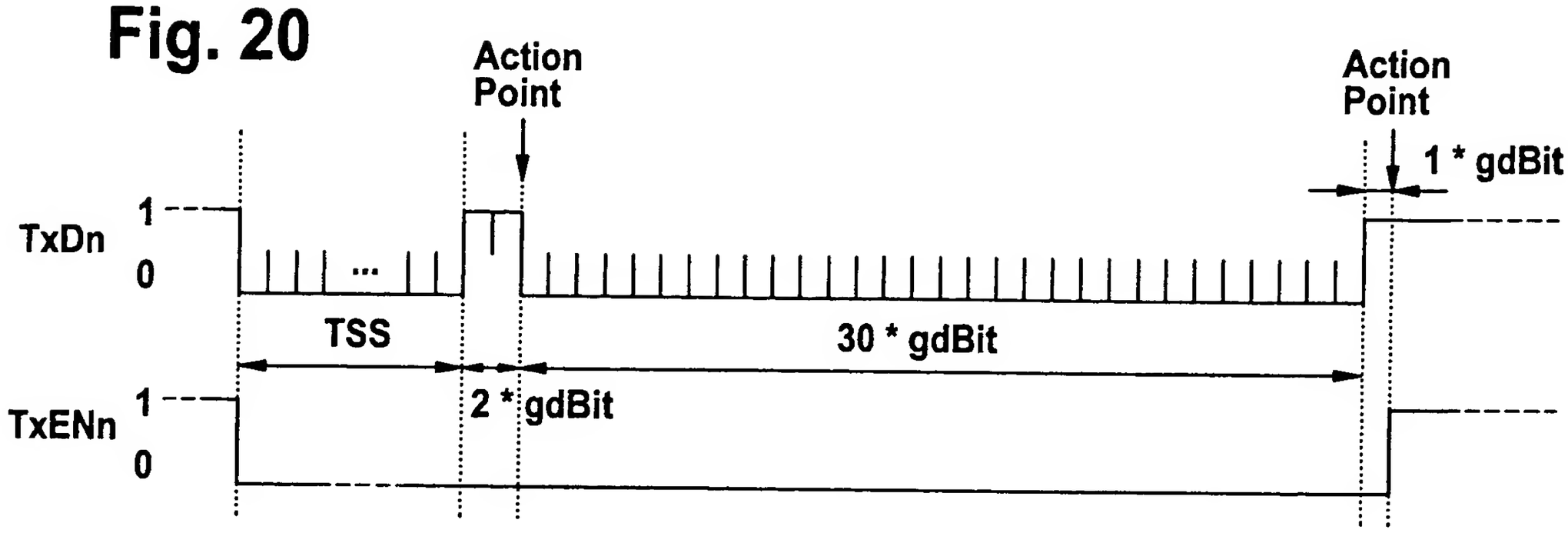


Fig. 21

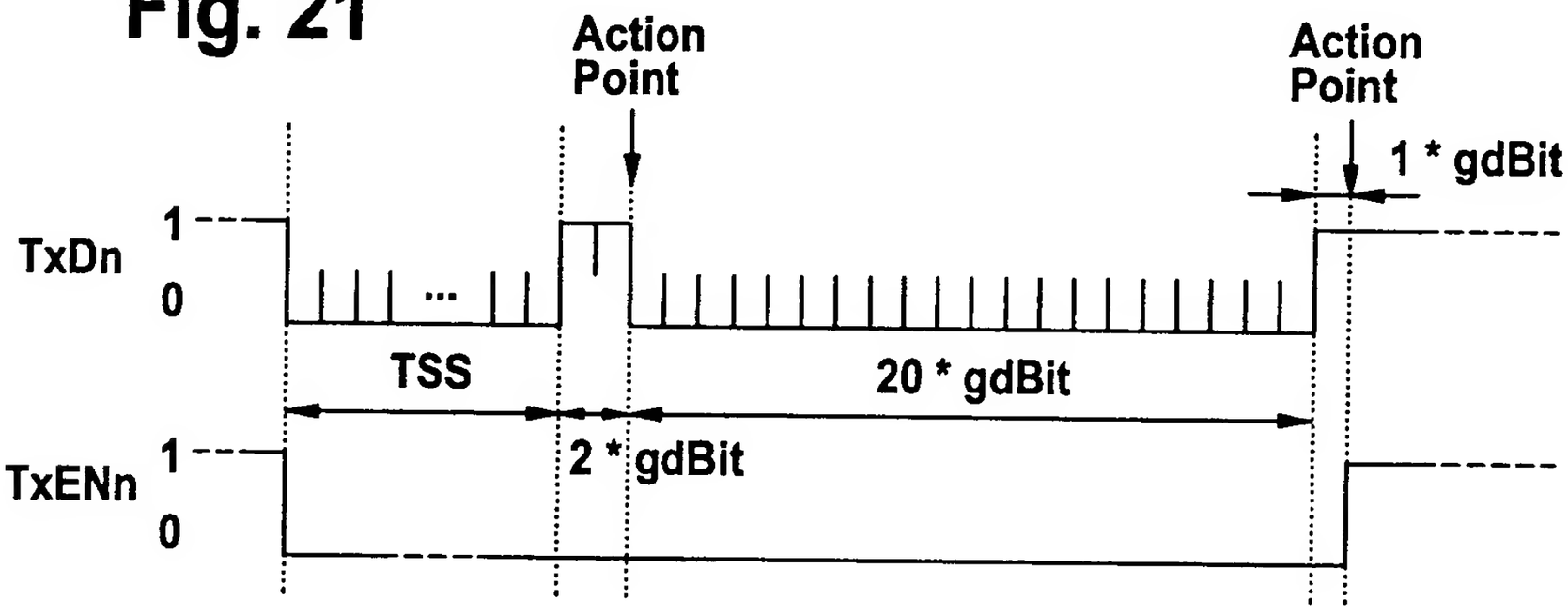


Fig. 22

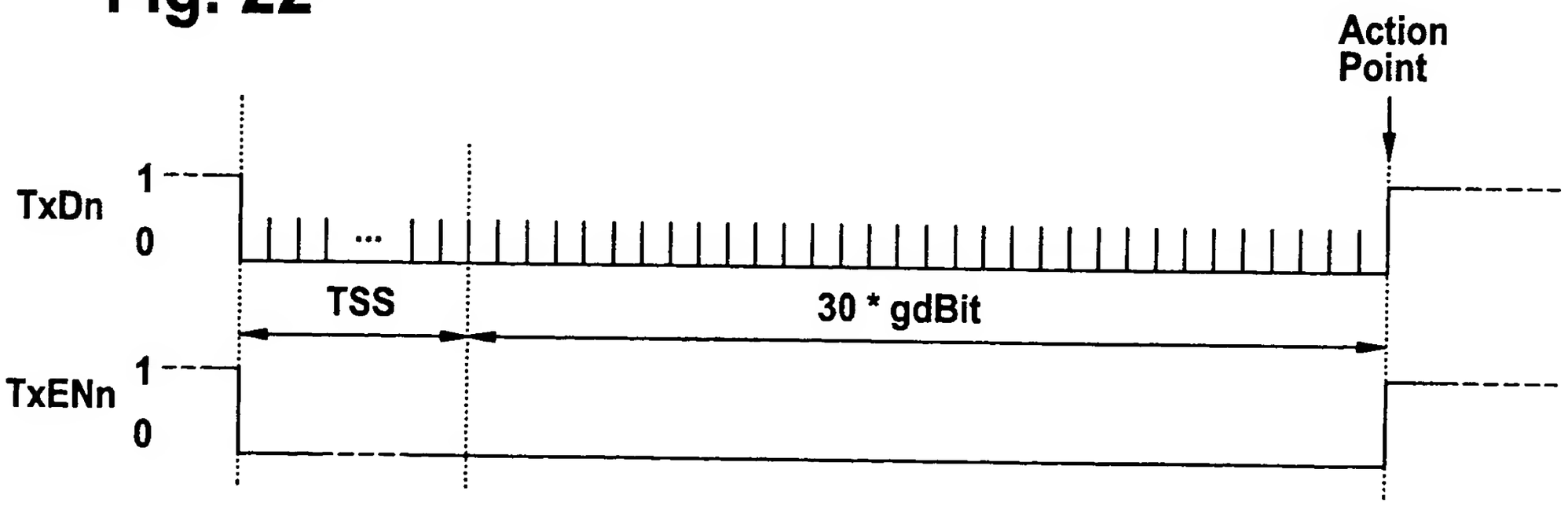


Fig. 23

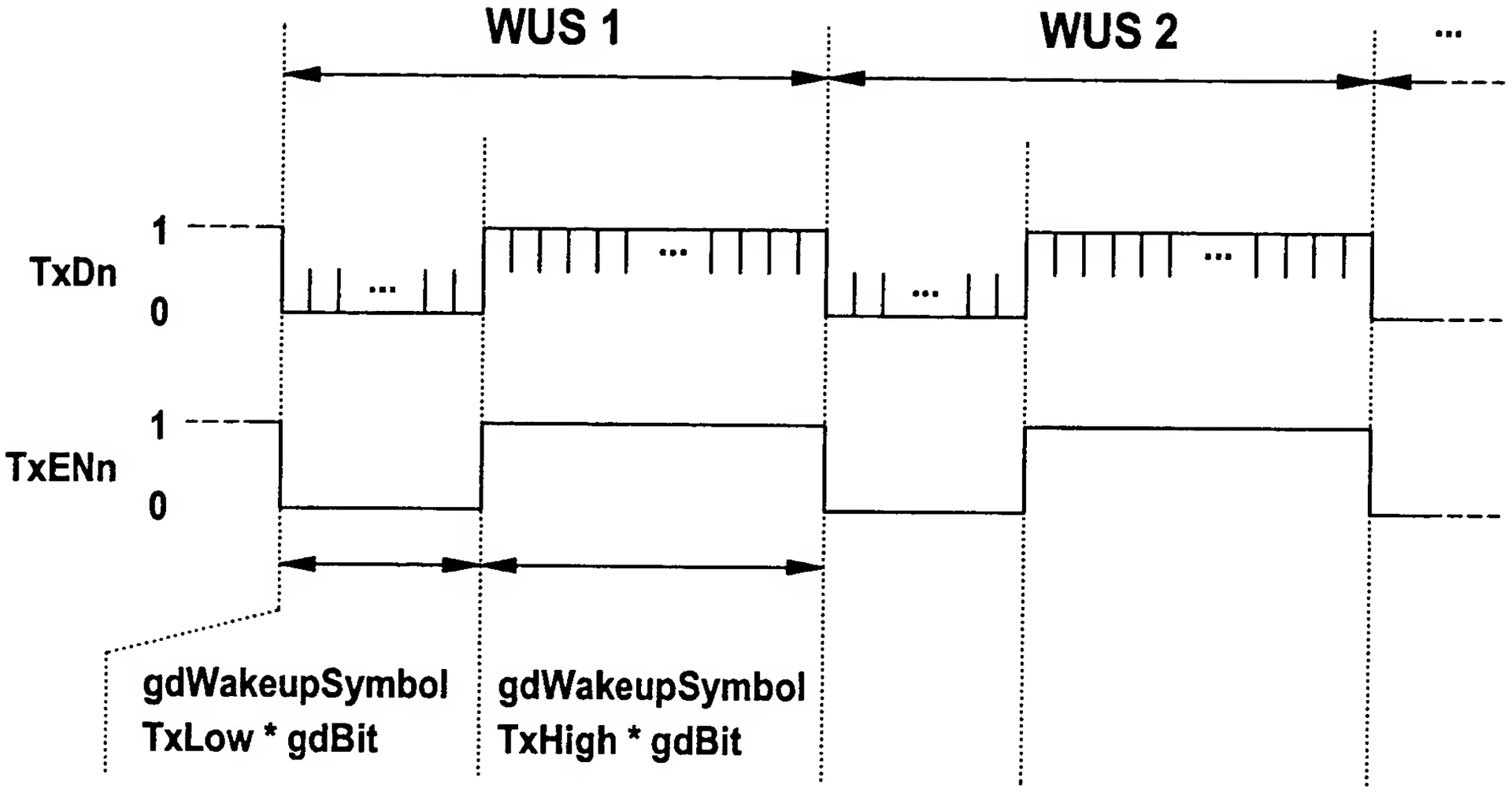

WUS 1
WUS 2
...
TxDn
1
0
TxENn
1
0
gdWakeupSymbol
TxLow * gdBit
gdWakeupSymbol
TxHigh * gdBit

Fig. 24

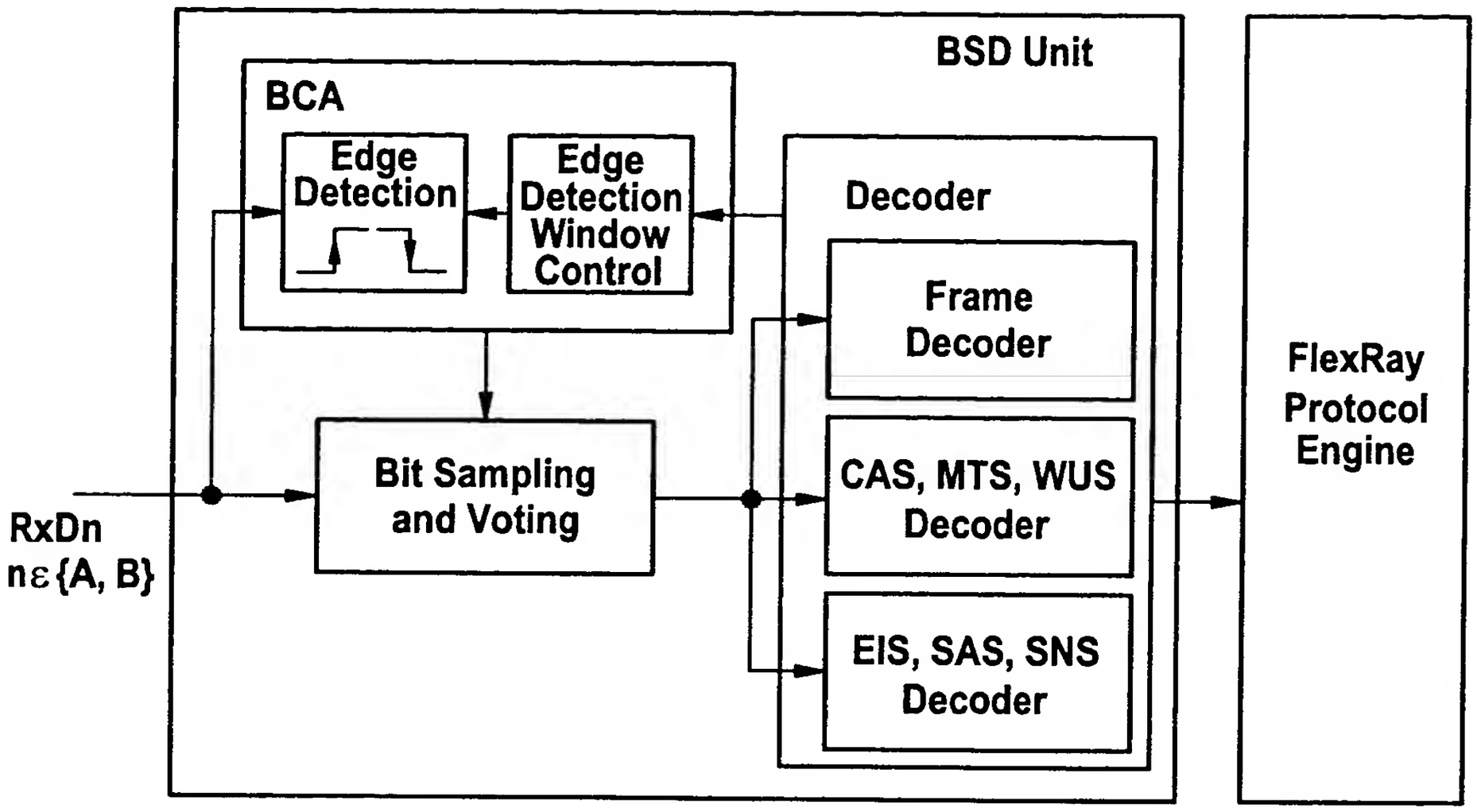

BSD Unit
BCA
Edge Detection
Edge Detection Window Control
Decoder
Frame Decoder
CAS, MTS, WUS Decoder
EIS, SAS, SNS Decoder
Bit Sampling and Voting
RxDn
nε{A, B}
FlexRay Protocol Engine

Fig. 25

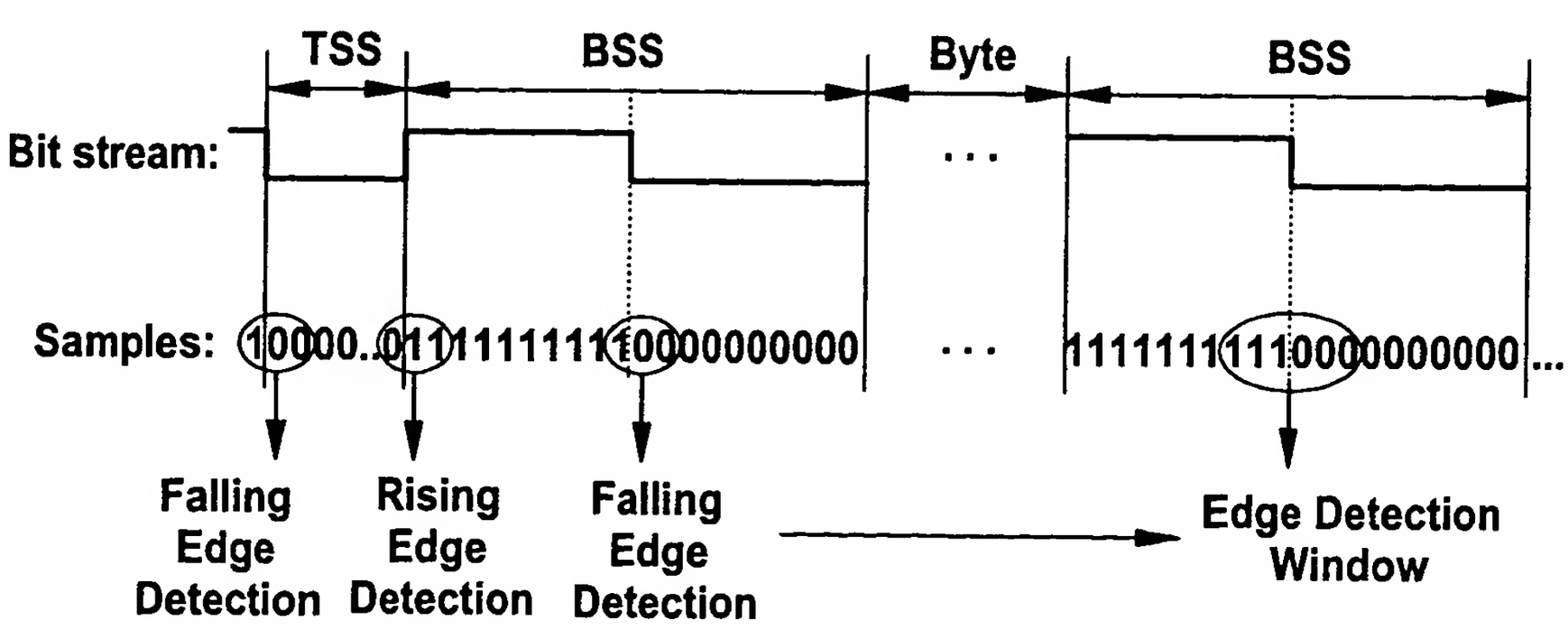

TSS
BSS
Byte
BSS
Bit stream:
Samples: 10000..011111111110000000000
1111111111000000000...
Falling Edge Detection
Rising Edge Detection
Falling Edge Detection
Edge Detection Window

Fig. 26

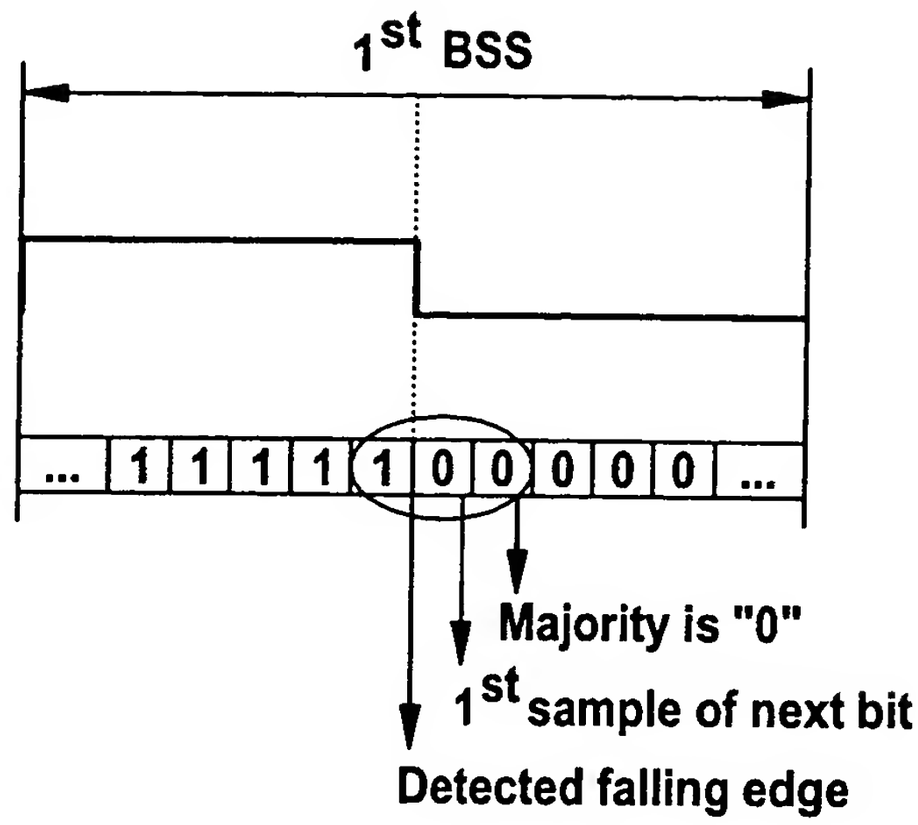

1st BSS
... 1 1 1 1 1 0 0 0 0 0 ...
Majority is "0"
1st sample of next bit
Detected falling edge

Fig. 27

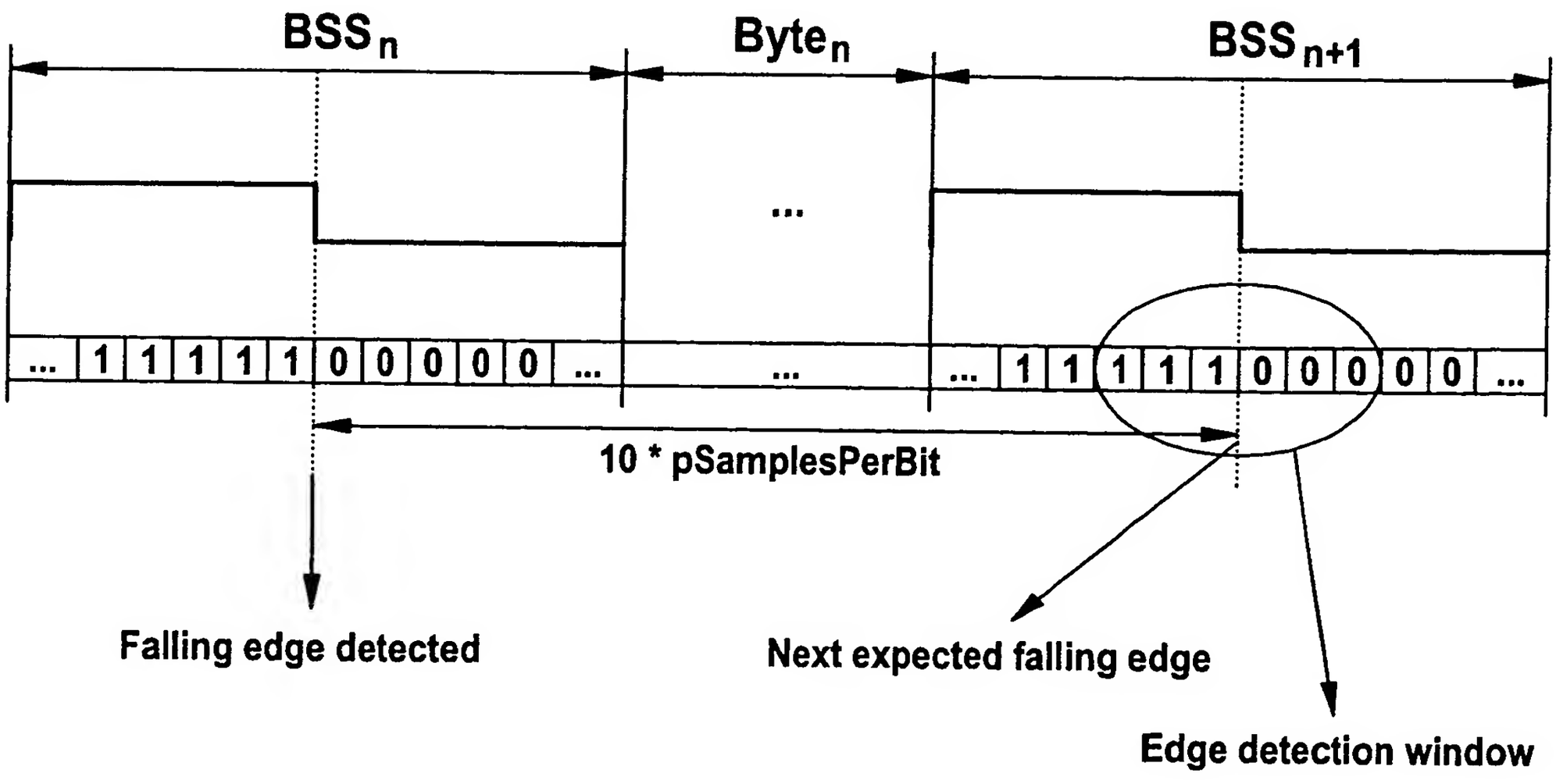

BSSn
Byten
BSSn+1
... 1 1 1 1 1 0 0 0 0 0 ...
... 1 1 1 1 1 0 0 0 0 0 ...
10 * pSamplesPerBit
Falling edge detected
Next expected falling edge
Edge detection window

Fig. 28

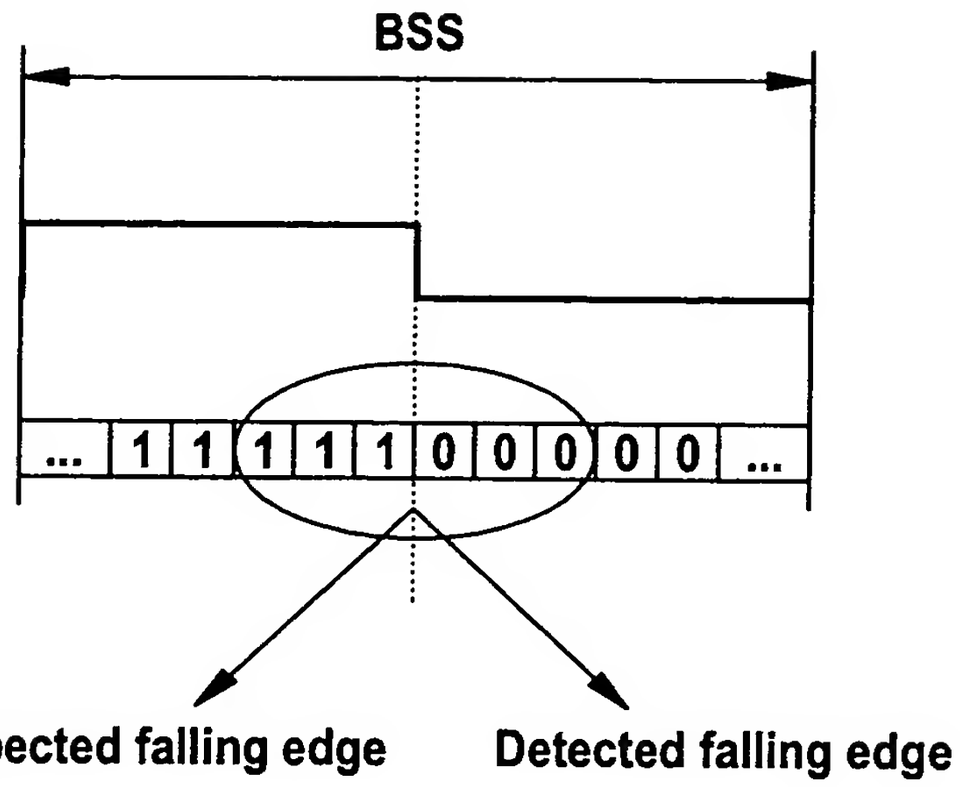


Fig. 29

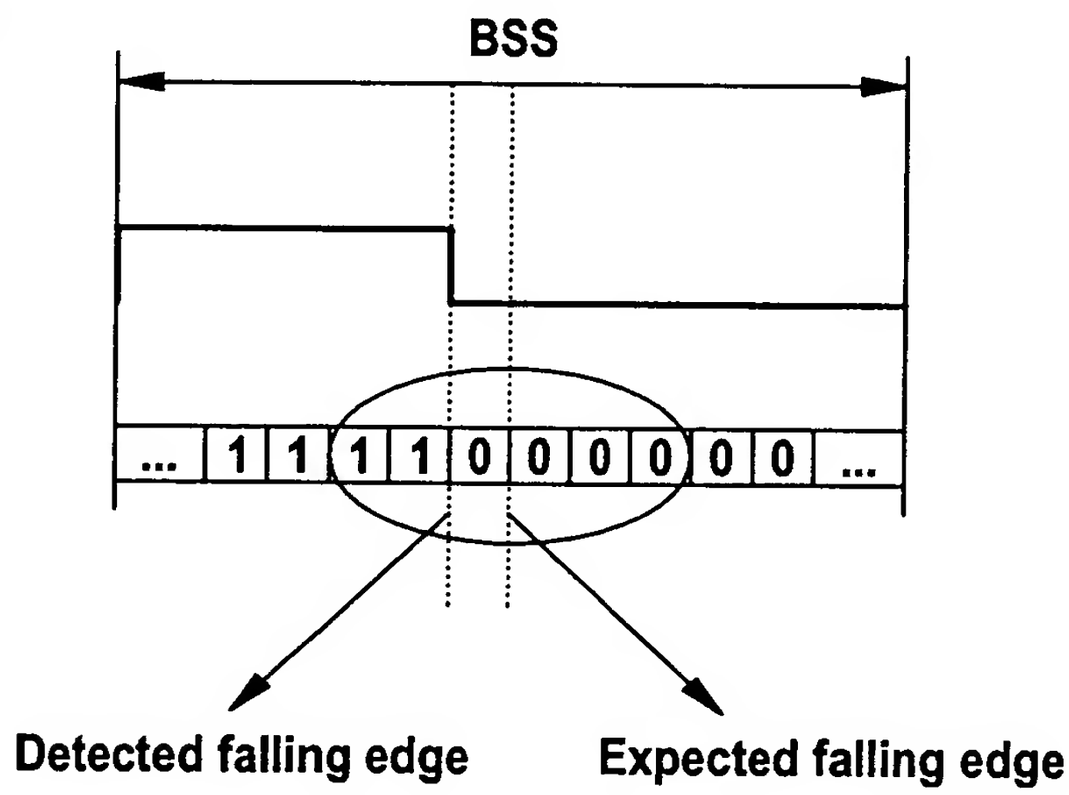


Fig. 30

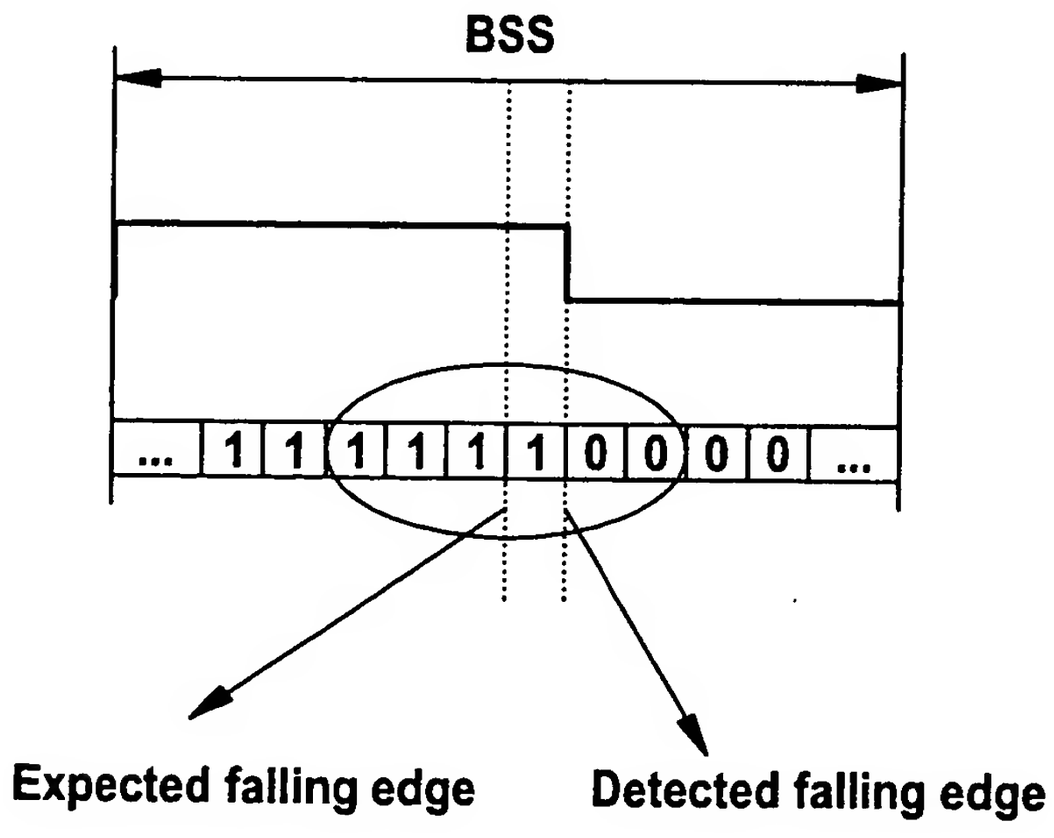


Fig. 31

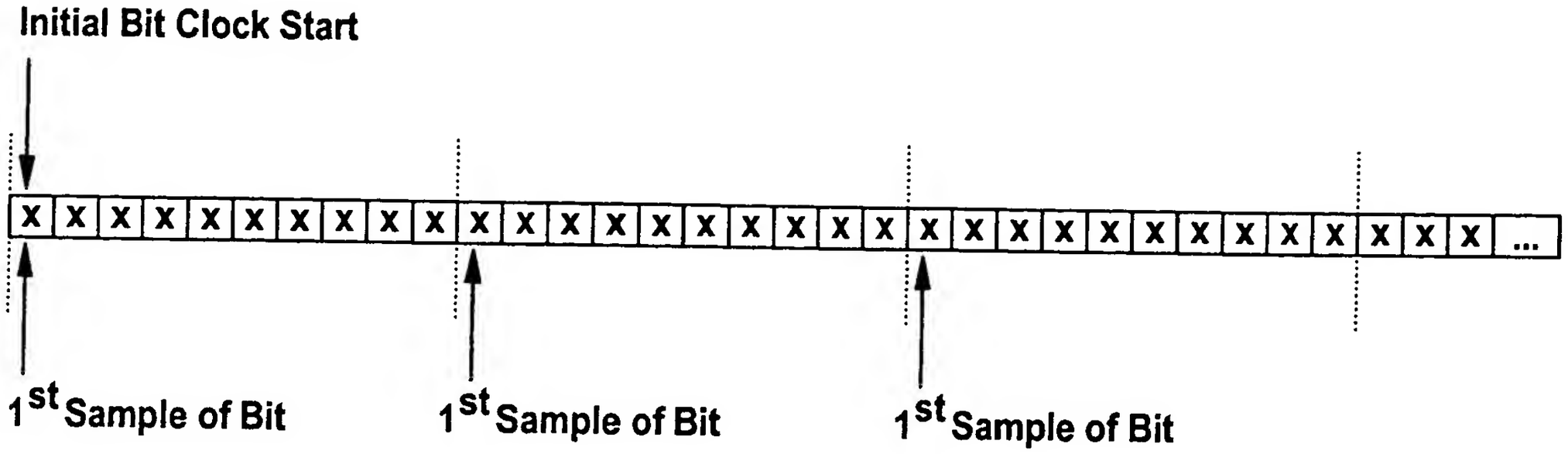


Fig. 32

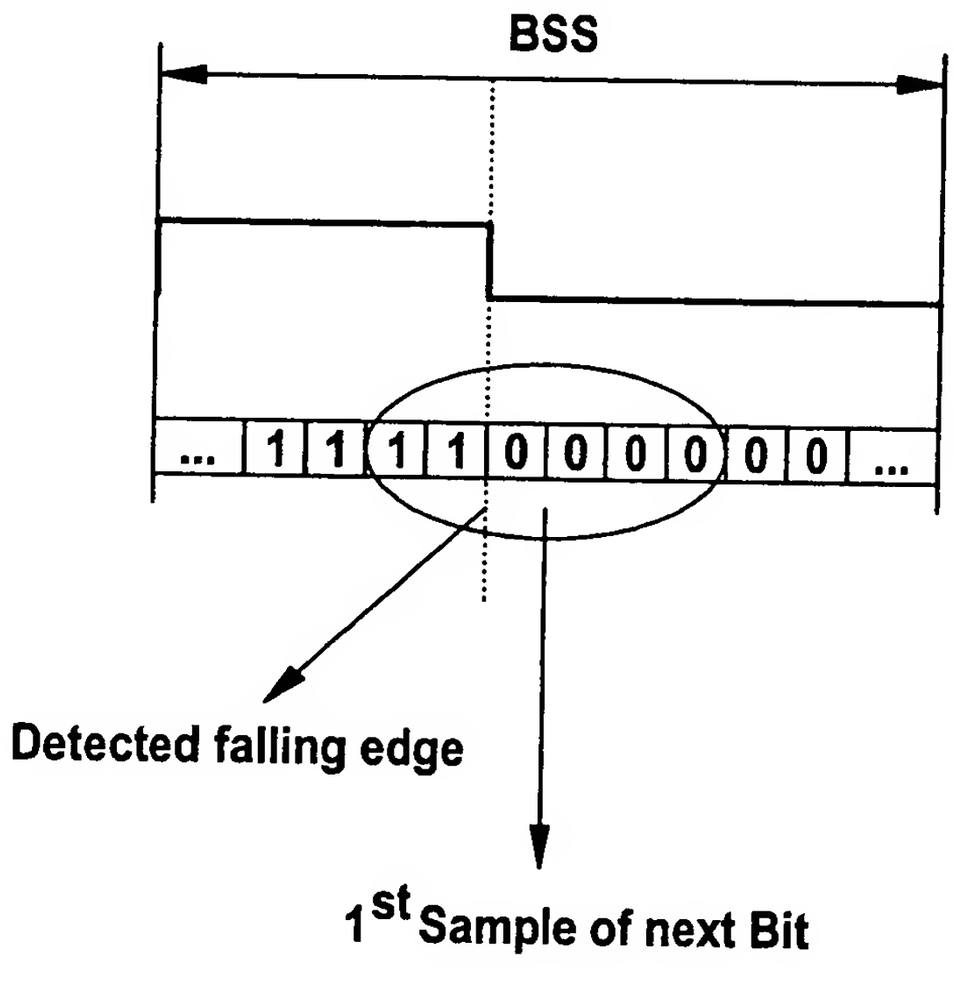


Fig. 33

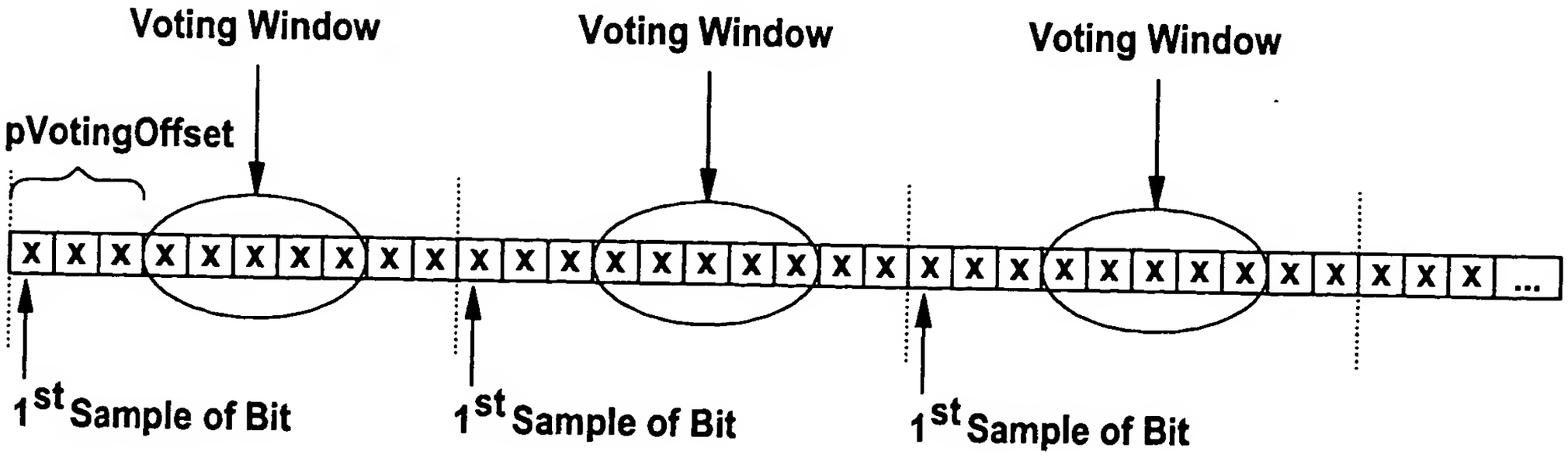


Fig. 34

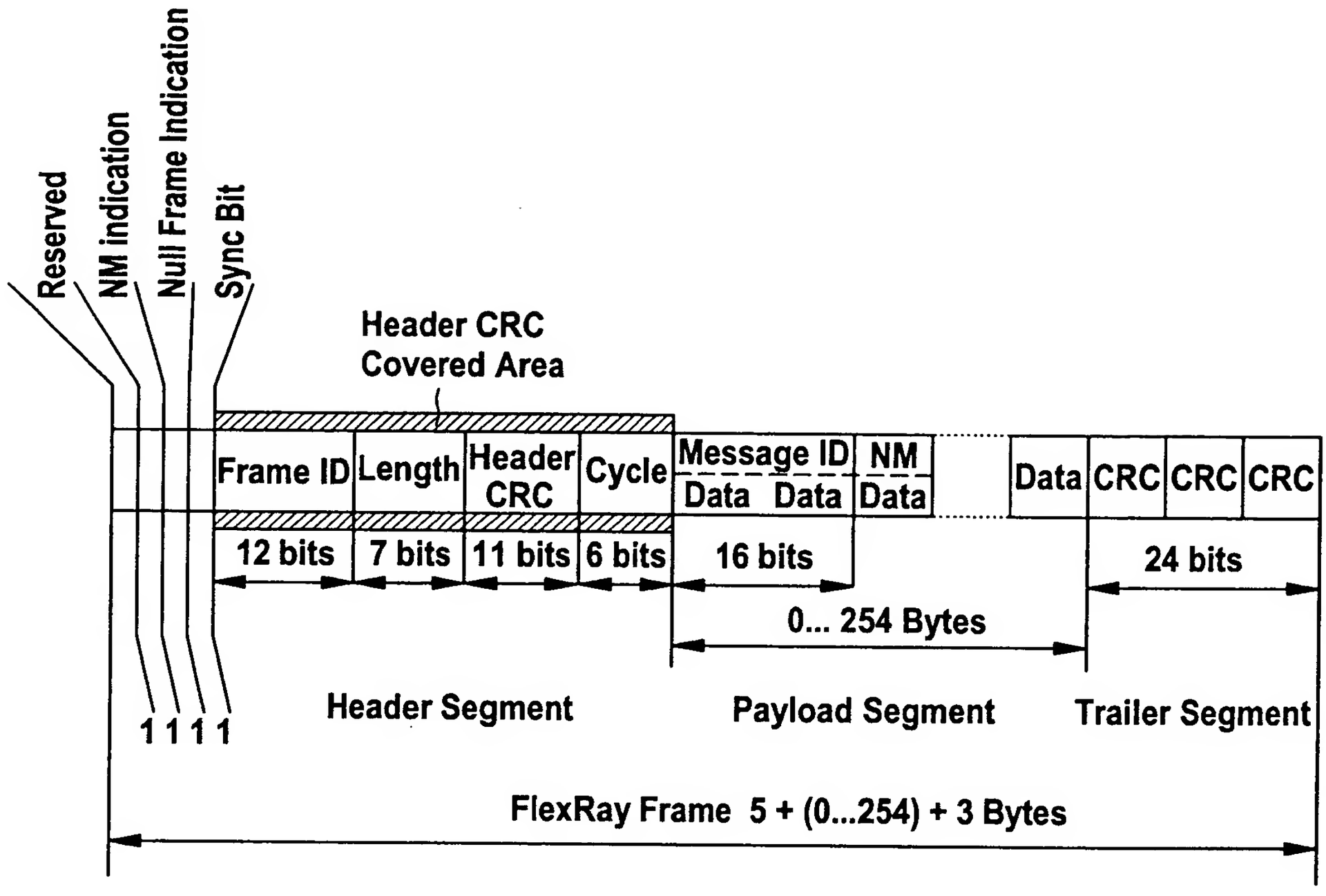

Reserved
NM indication
Null Frame Indication
Sync Bit
Header CRC
Covered Area
Frame ID
Length
Header CRC
Cycle
Message ID
Data
NM
Data
Data
CRC
CRC
CRC
12 bits
7 bits
11 bits
6 bits
16 bits
24 bits
0... 254 Bytes
1 1 1 1
Header Segment
Payload Segment
Trailer Segment
FlexRay Frame 5 + (0...254) + 3 Bytes

Fig. 35

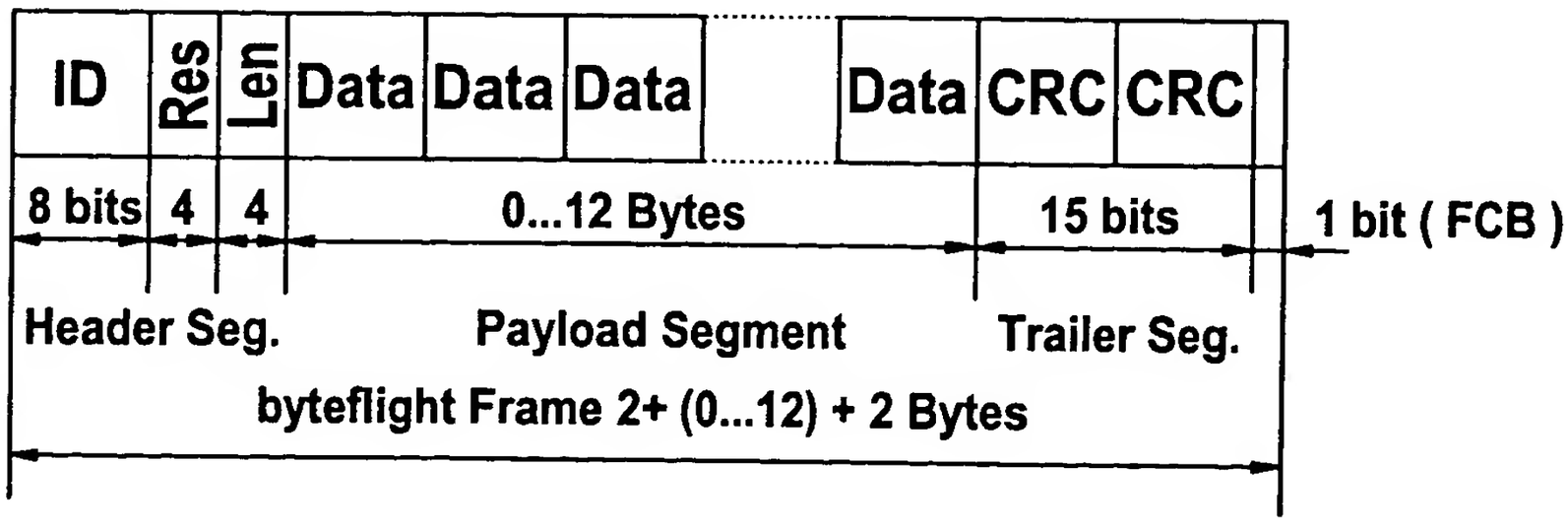

ID
Res
Len
Data
Data
Data
Data
CRC
CRC
8 bits
4
4
0...12 Bytes
15 bits
1 bit (FCB)
Header Seg.
Payload Segment
Trailer Seg.
byteflight Frame 2+ (0...12) + 2 Bytes

Fig. 36

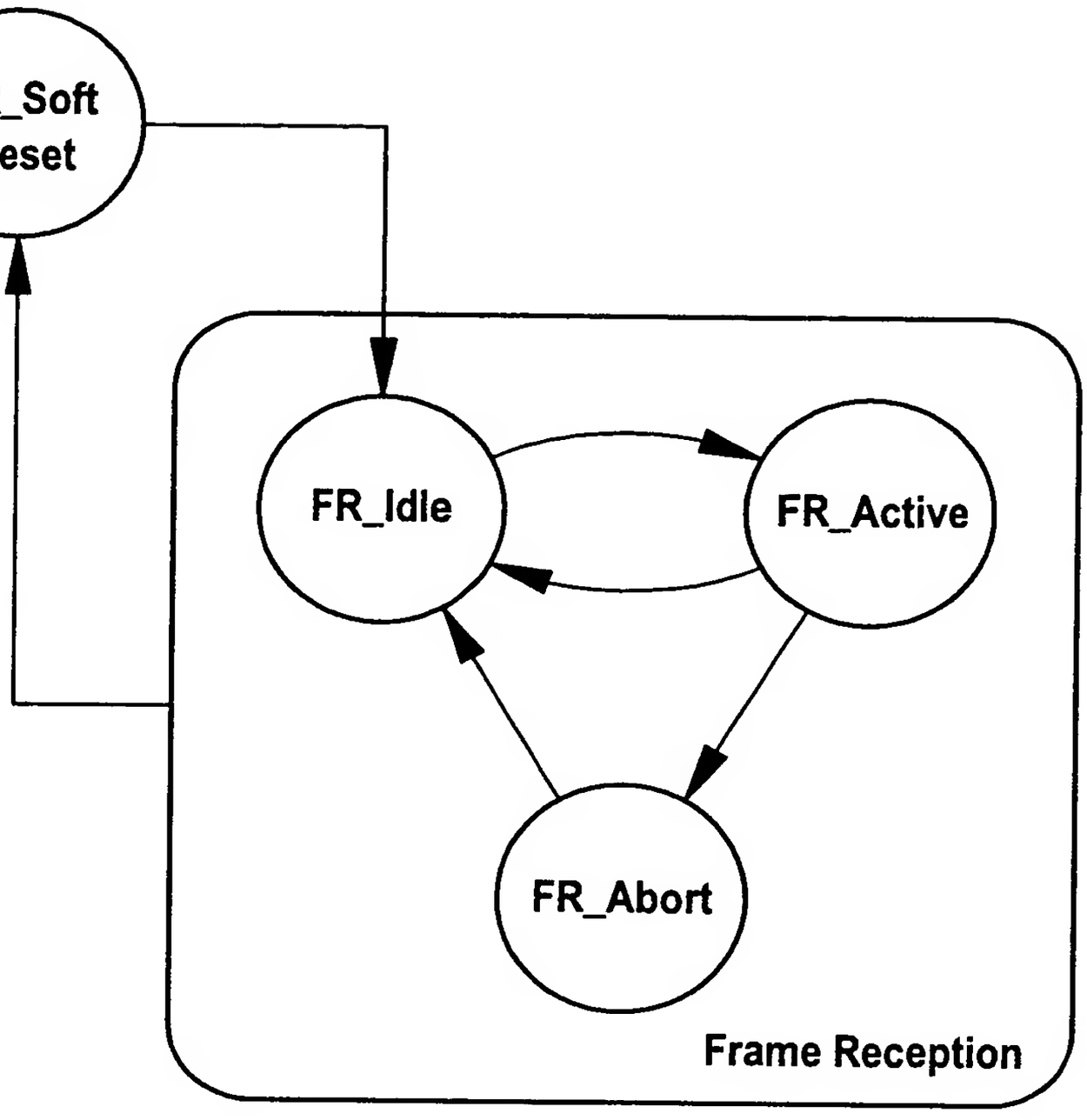

FR_Soft
Reset
FR_Idle
FR_Active
FR_Abort
Frame Reception

Fig. 37

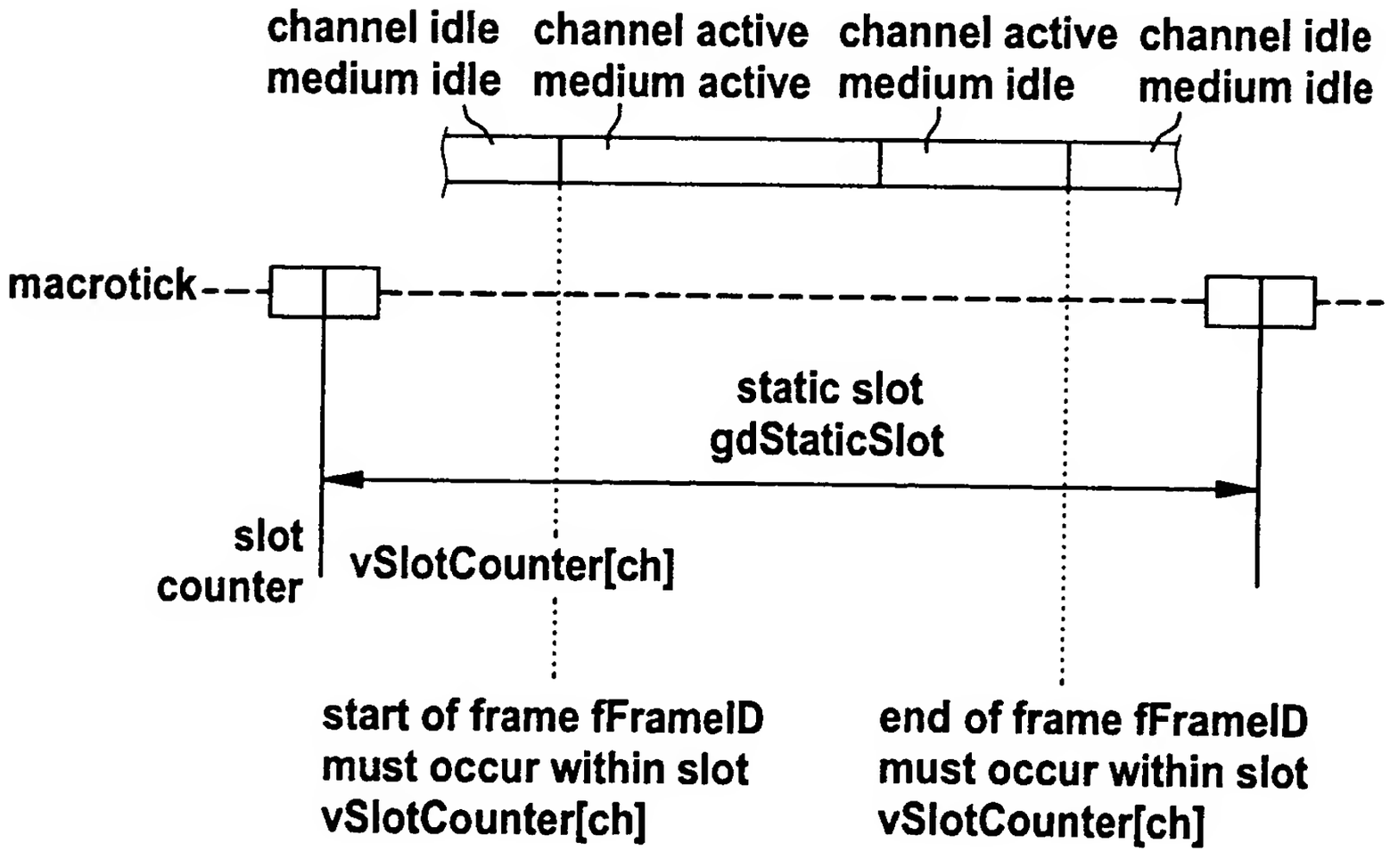

channel idle
channel active
channel active
channel idle
medium idle
medium active
medium idle
medium idle
macrotick
static slot
gdStaticSlot
slot
counter
vSlotCounter[ch]
start of frame fFrameID
must occur within slot
vSlotCounter[ch]
end of frame fFrameID
must occur within slot
vSlotCounter[ch]

Fig. 38

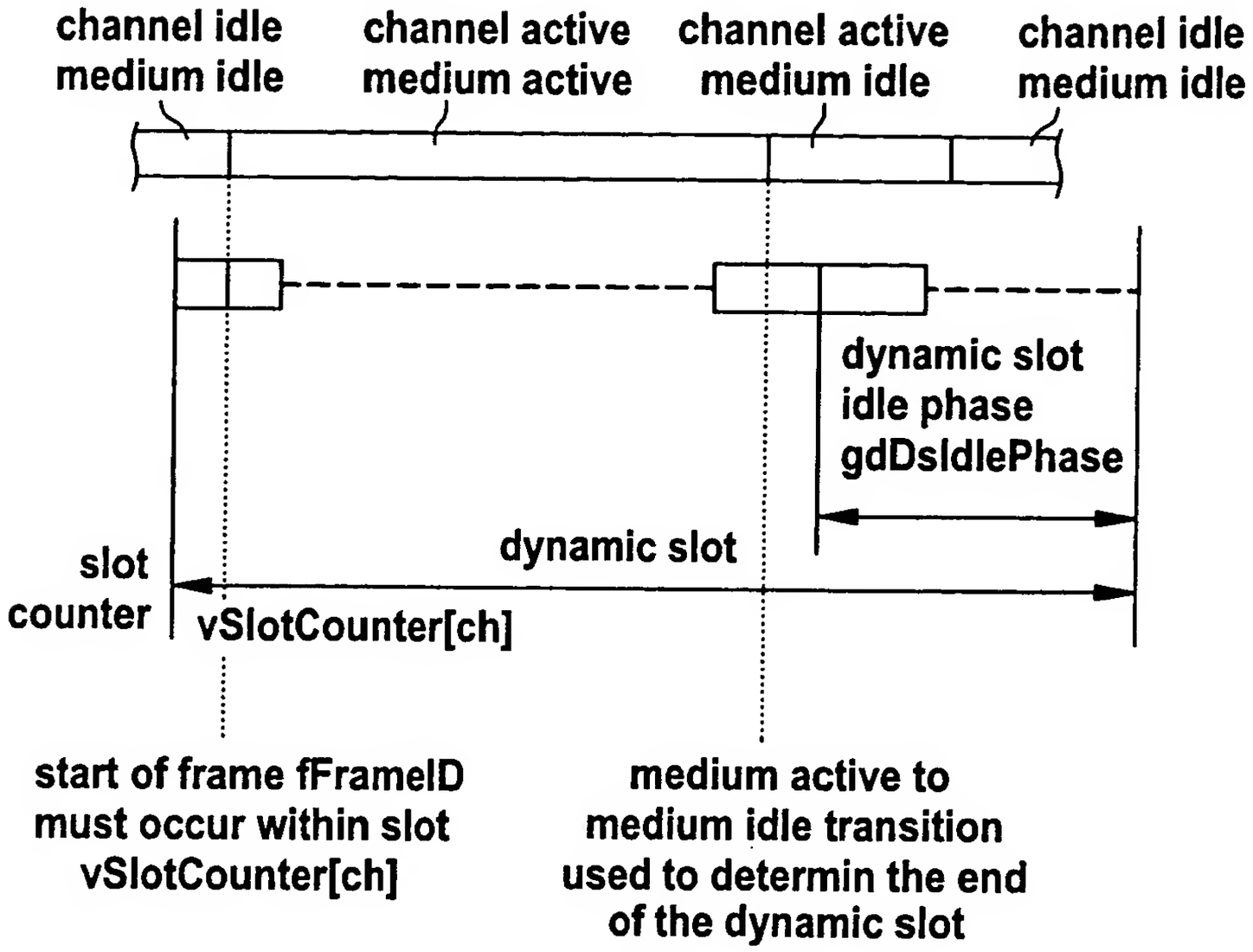
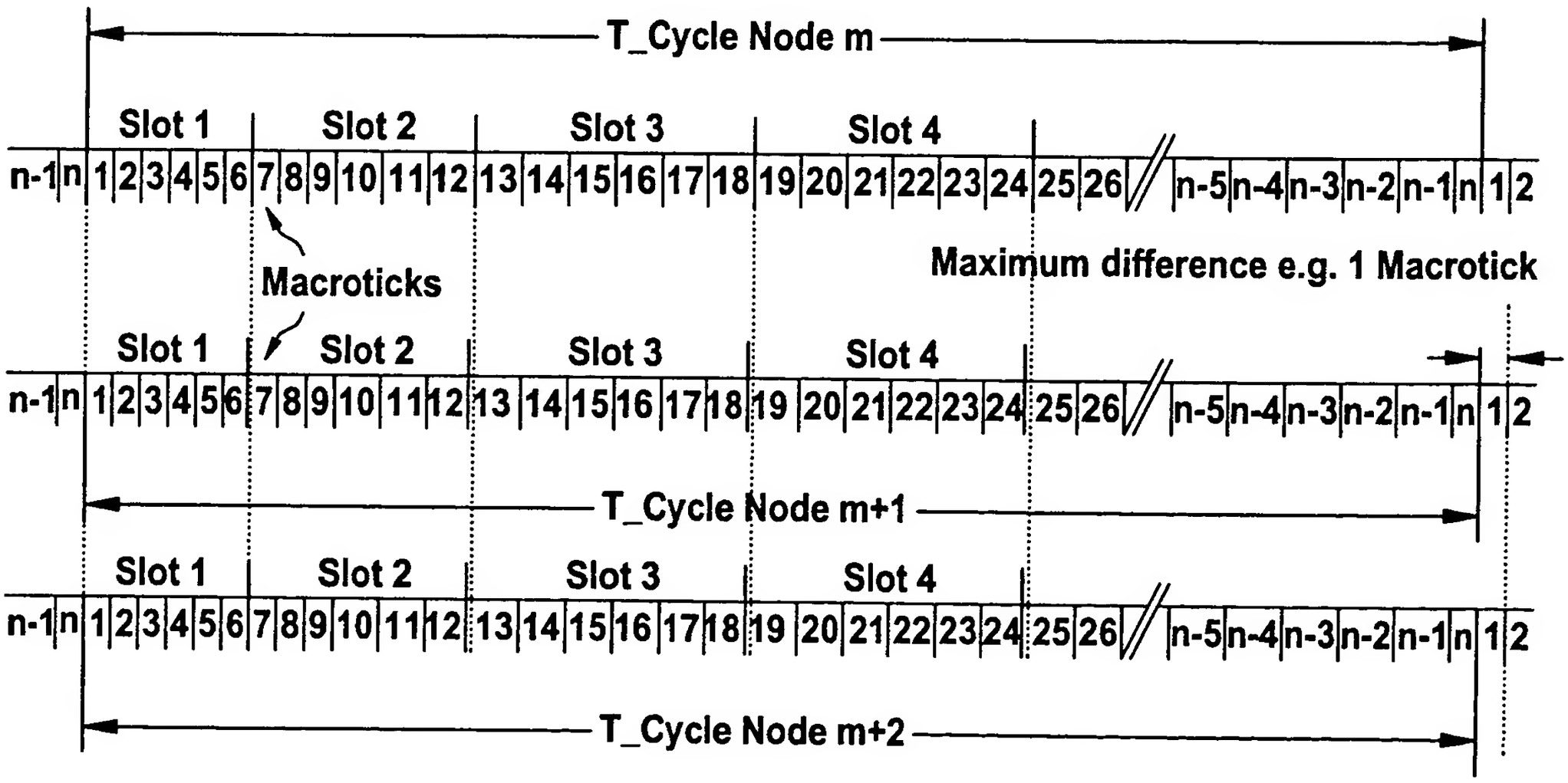


Fig. 39



n = number of Macroticks per cycle

Fig. 40

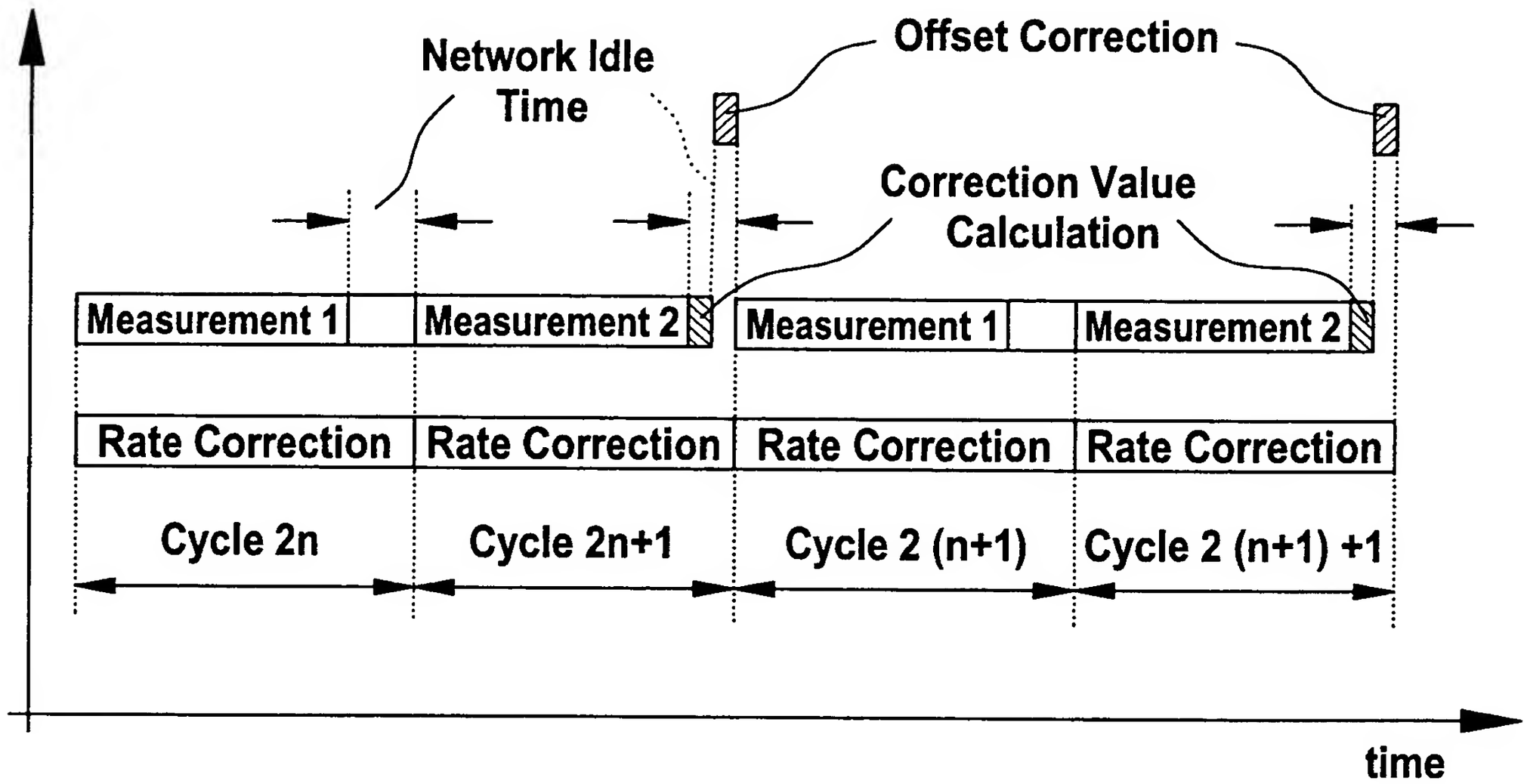
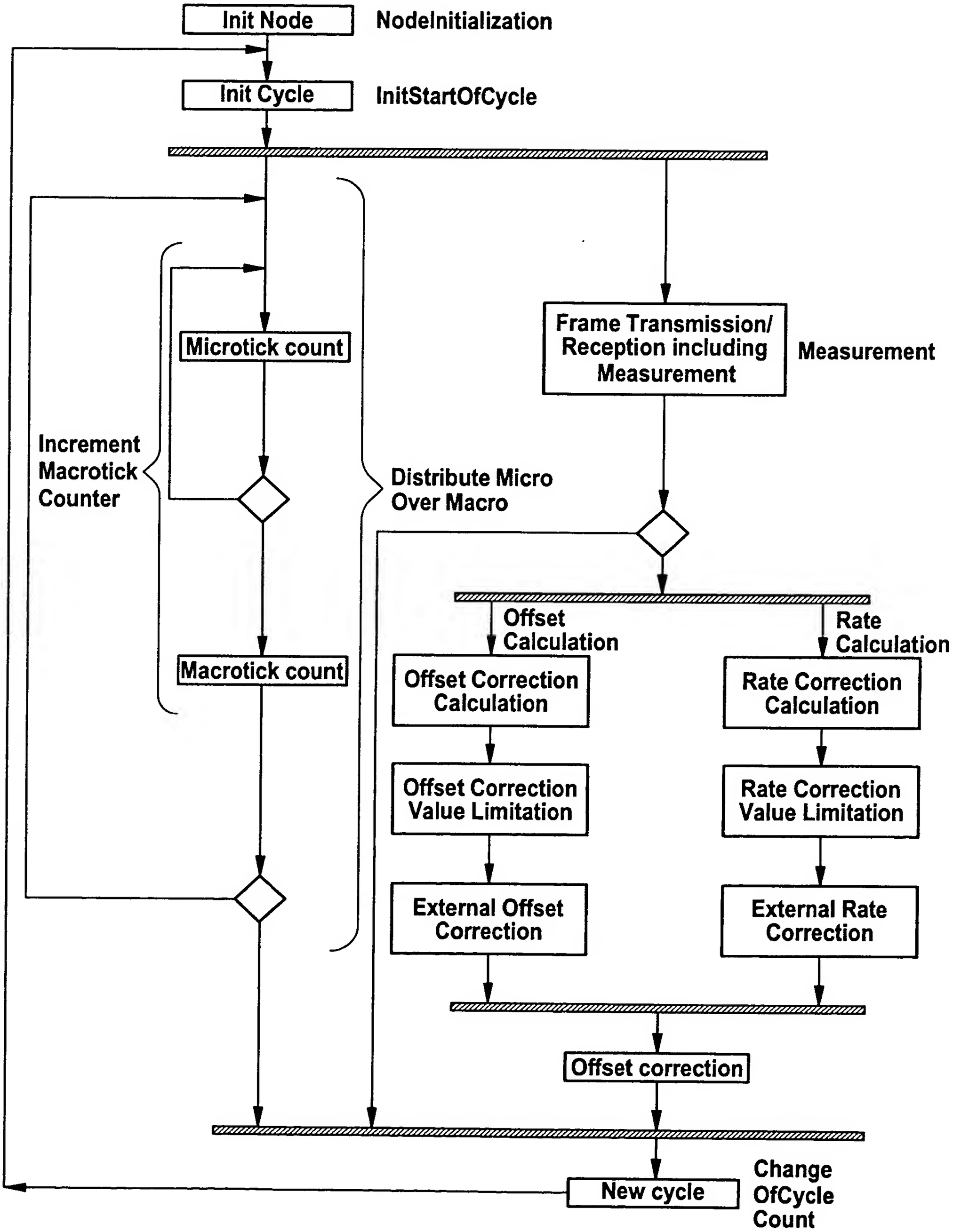

Network Idle Time
Offset Correction
Correction Value Calculation
Measurement 1
Measurement 2
Measurement 1
Measurement 2
Rate Correction
Rate Correction
Rate Correction
Rate Correction
Cycle 2n
Cycle 2n+1
Cycle 2 (n+1)
Cycle 2 (n+1) +1
time

Fig. 41



06071457EP

Fig. 42

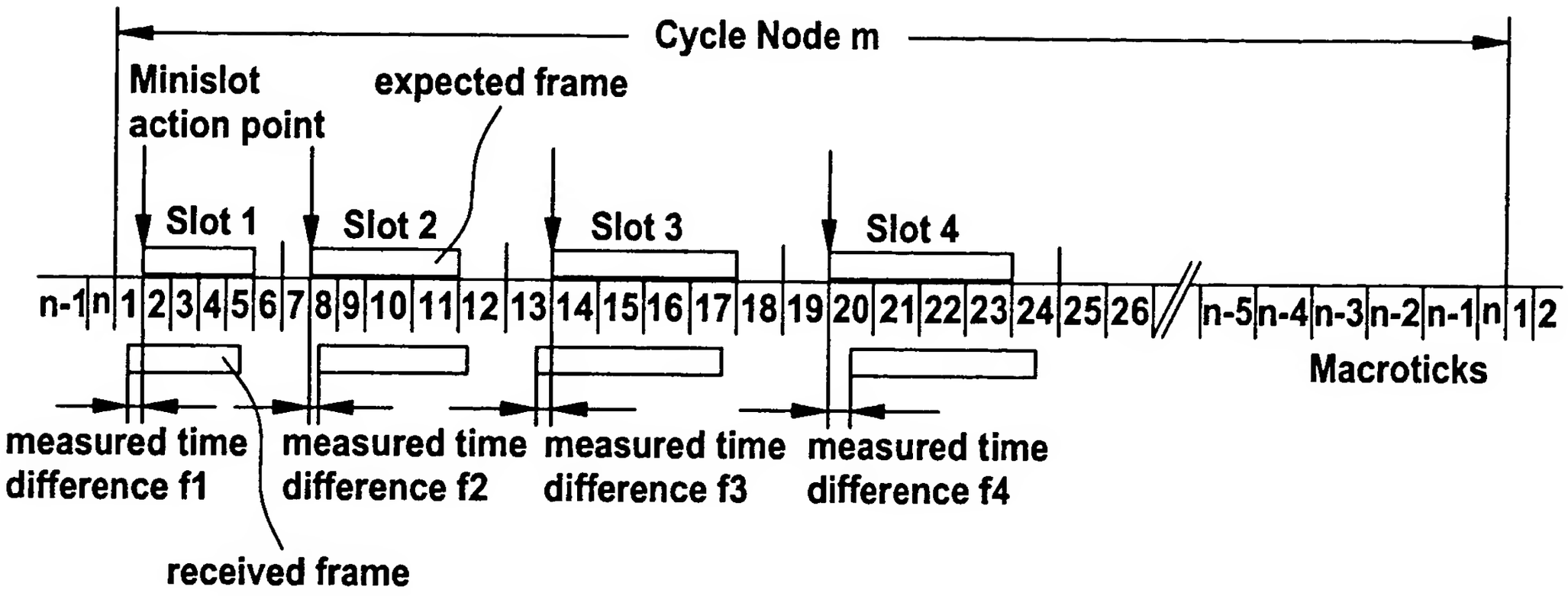


Fig. 43

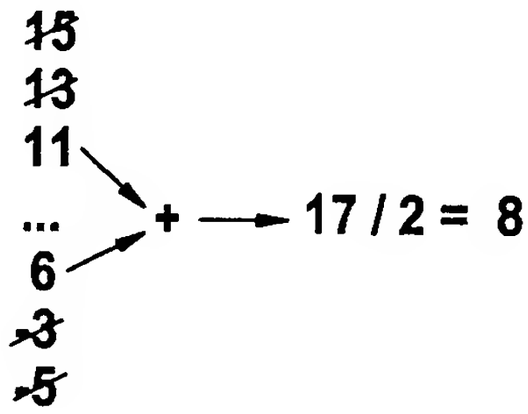


Fig. 44

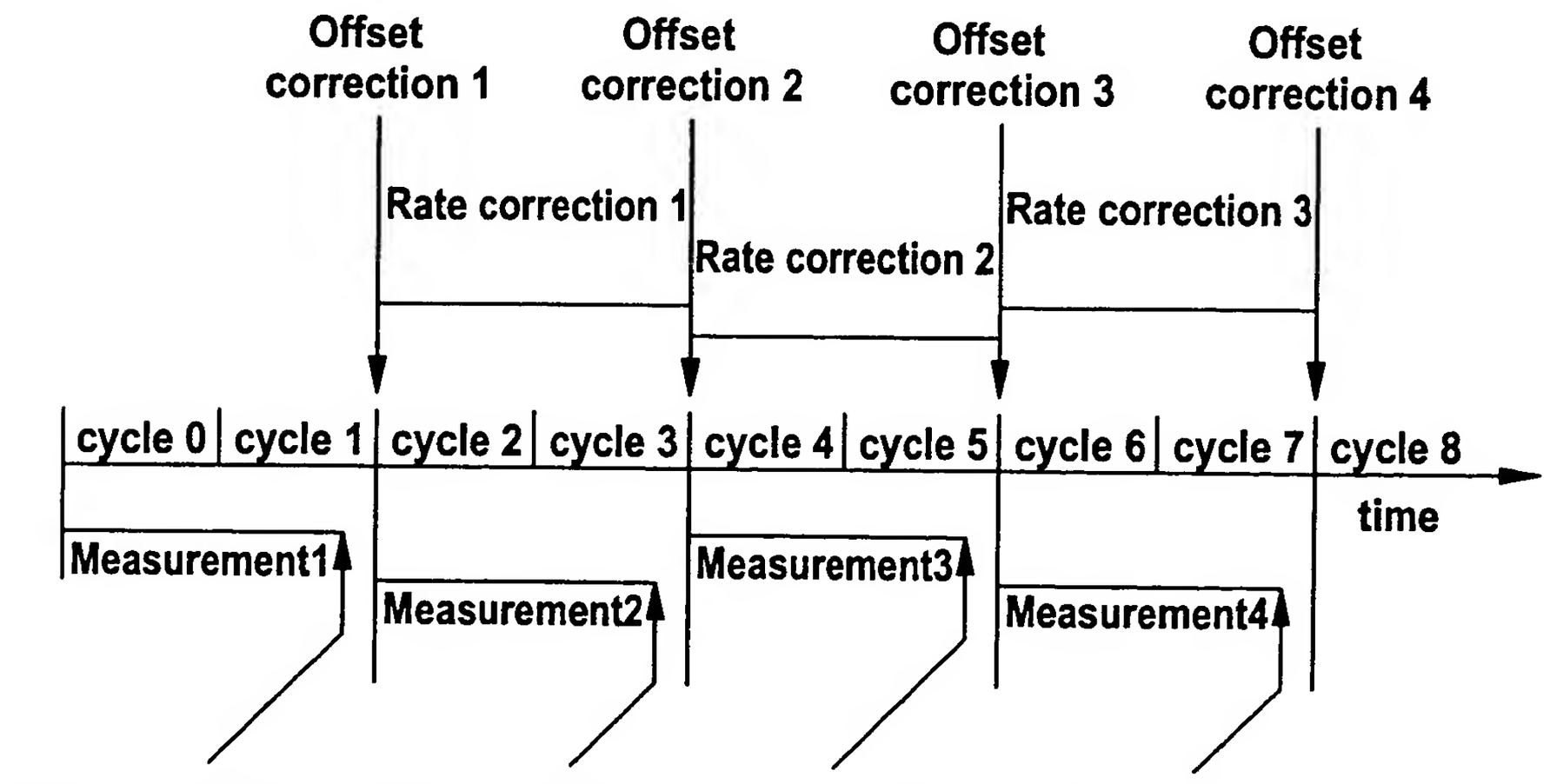


Fig. 45

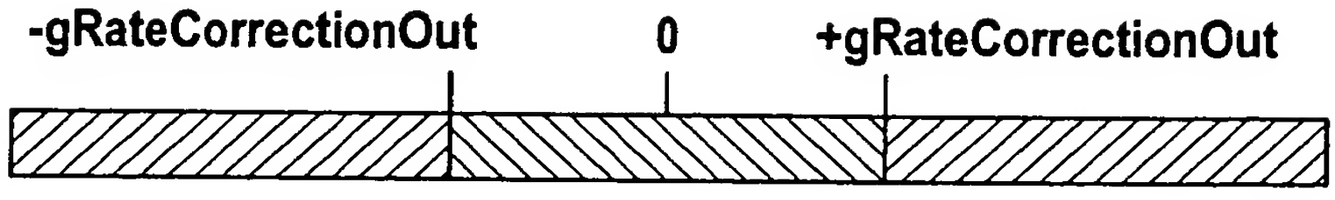


Fig. 46

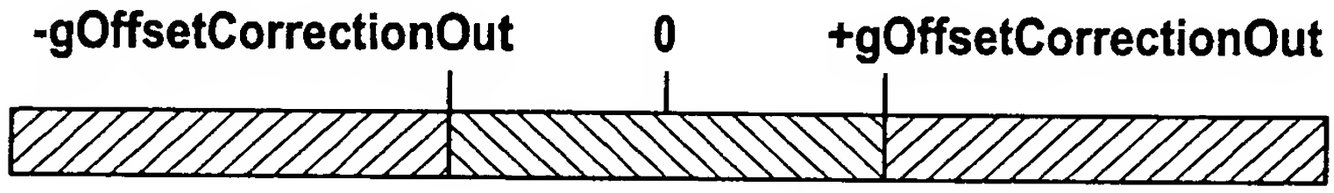


Fig. 47

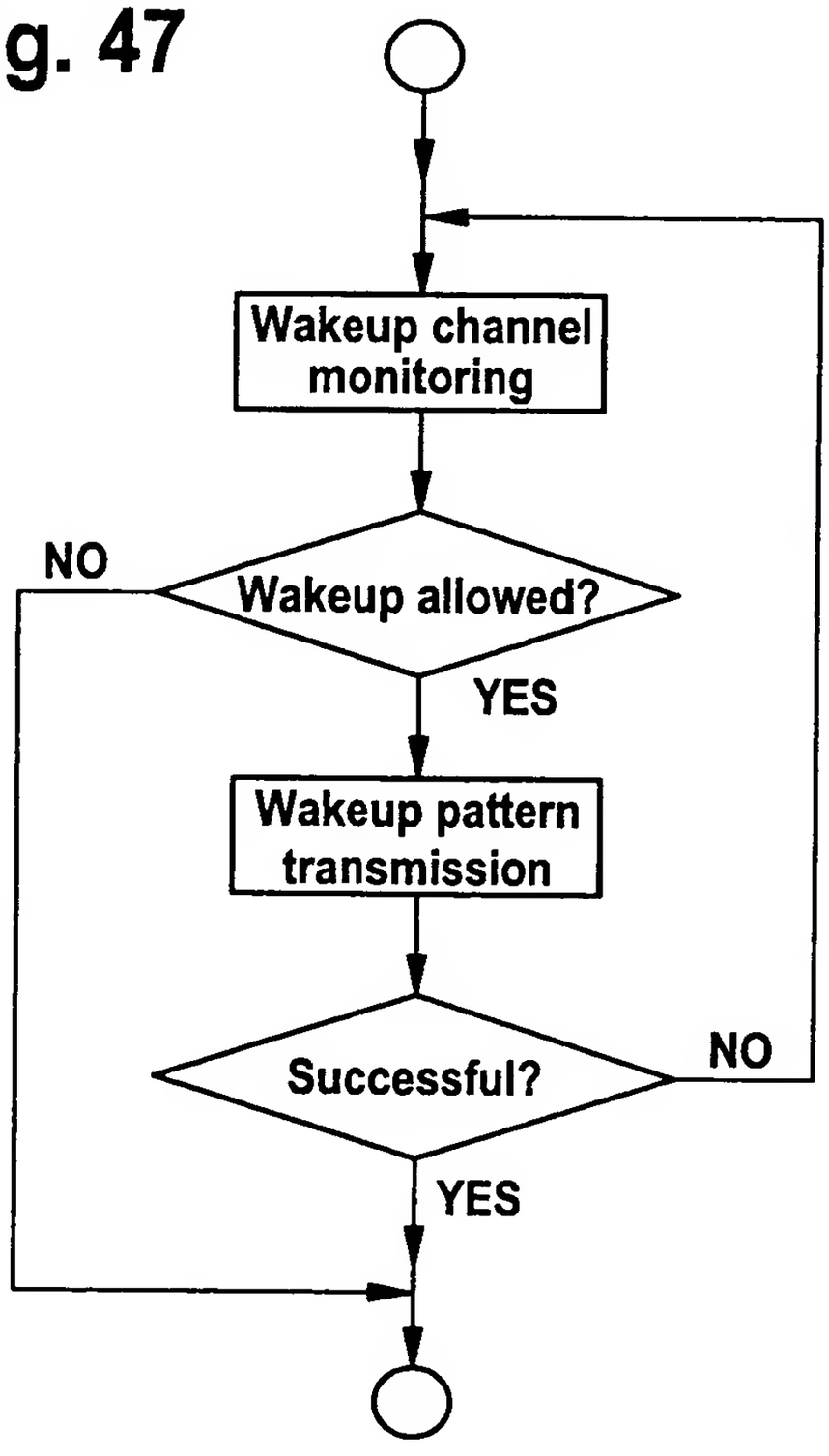


Fig. 48

Wakeup channel monitoring

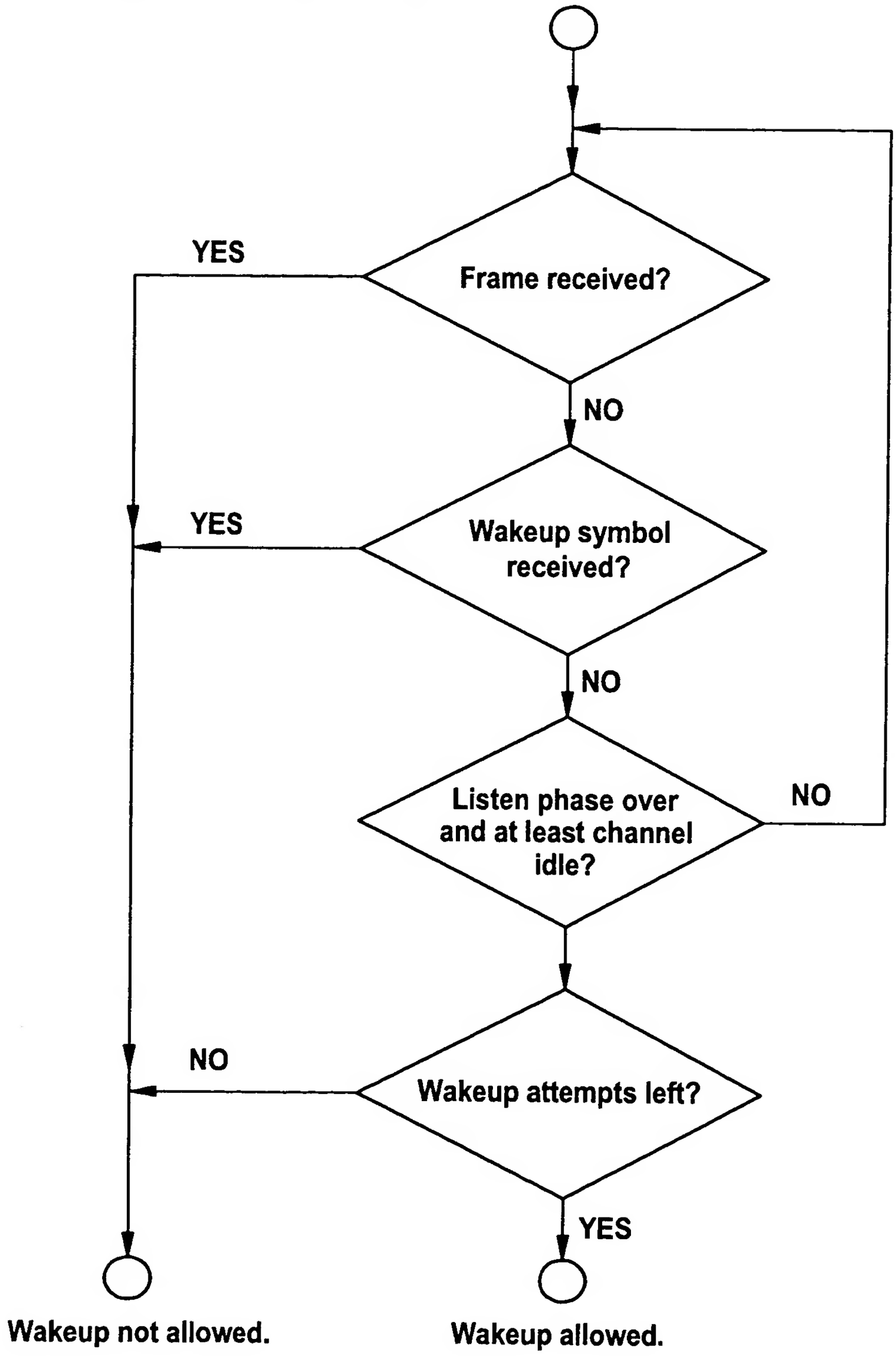


Fig. 49

Wakeup pattern transmission

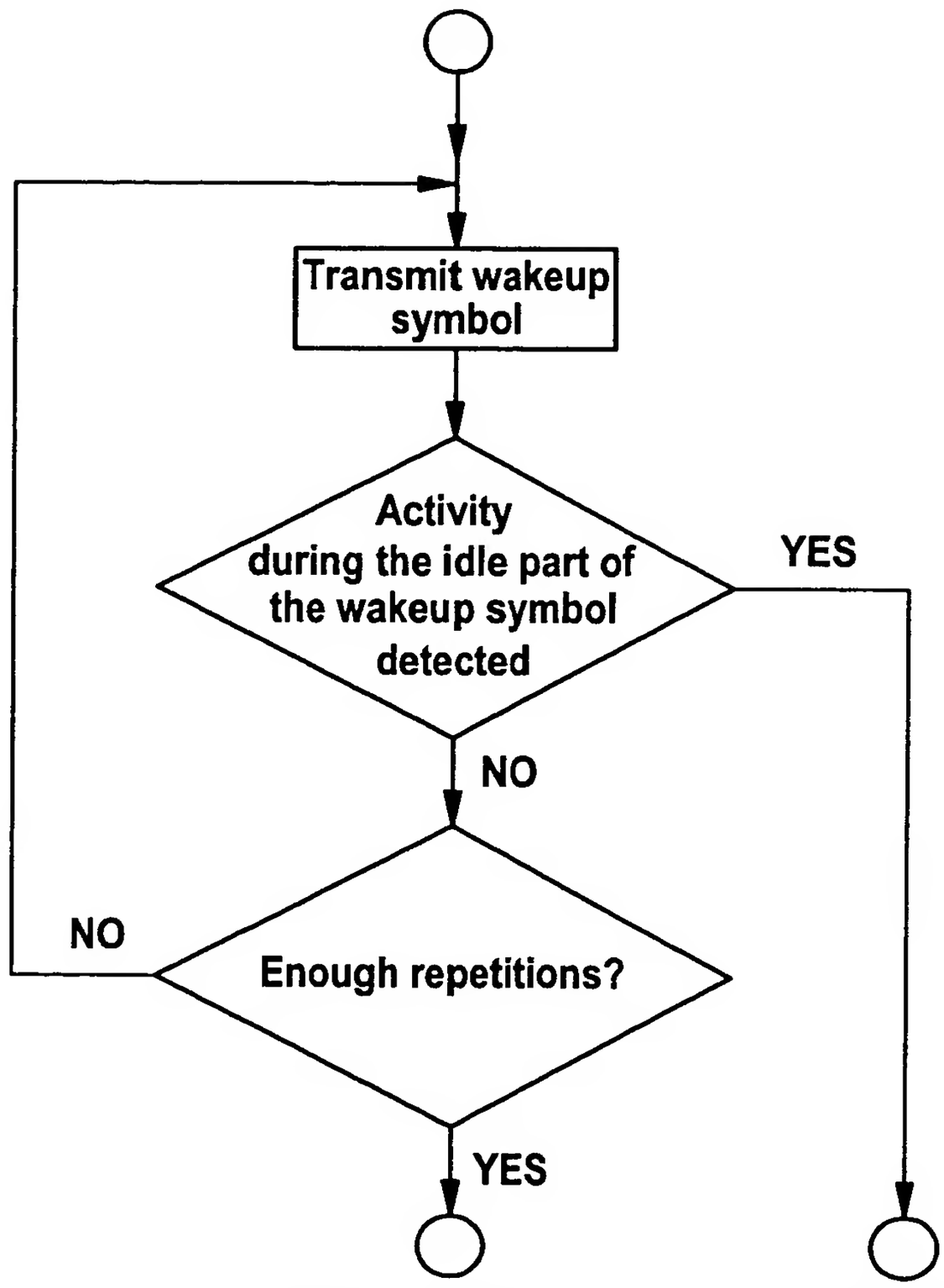


Fig. 50

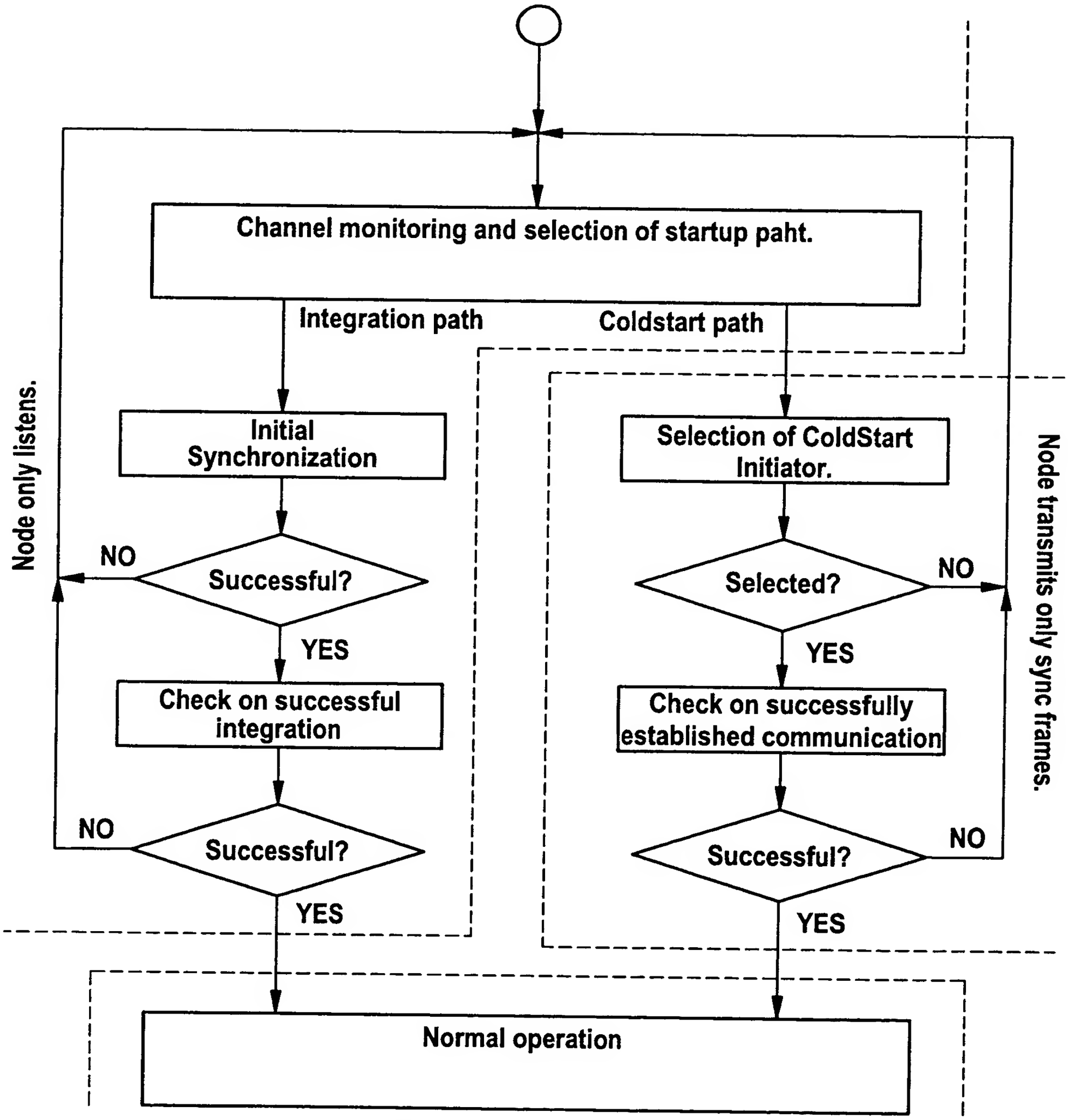

Channel monitoring and selection of startup paht.
Integration path
Coldstart path
Initial Synchronization
Successful?
NO
YES
Check on successful integration
Successful?
NO
YES
Node only listens.
Selection of ColdStart Initiator.
Selected?
NO
YES
Check on successfully established communication
Successful?
NO
YES
Node transmits only sync frames.
Normal operation
Node communicates normally.

Fig. 51

Channel monitoring and selection of startup path.

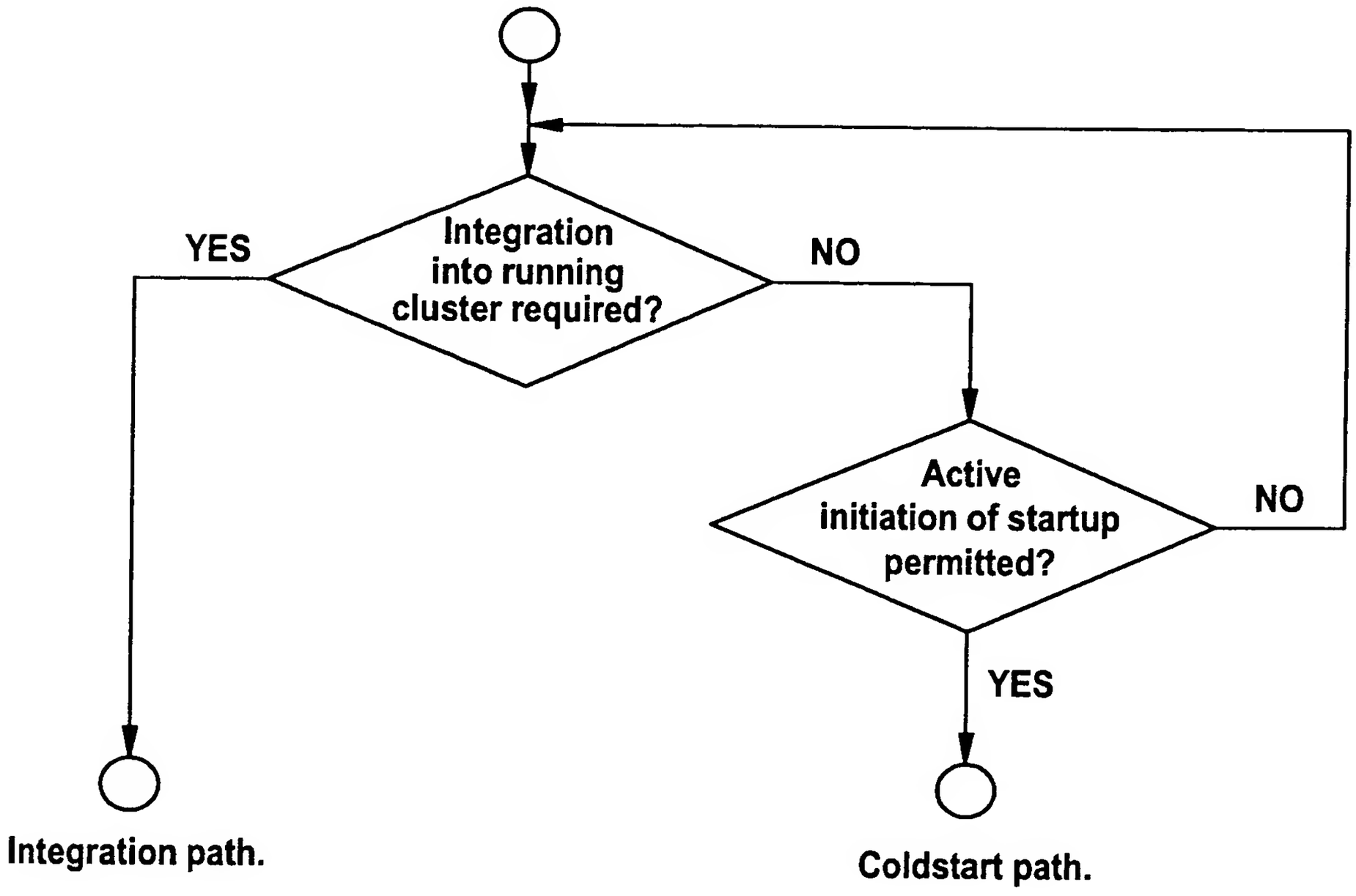


Fig. 52

Selection of ColdStart Initiator

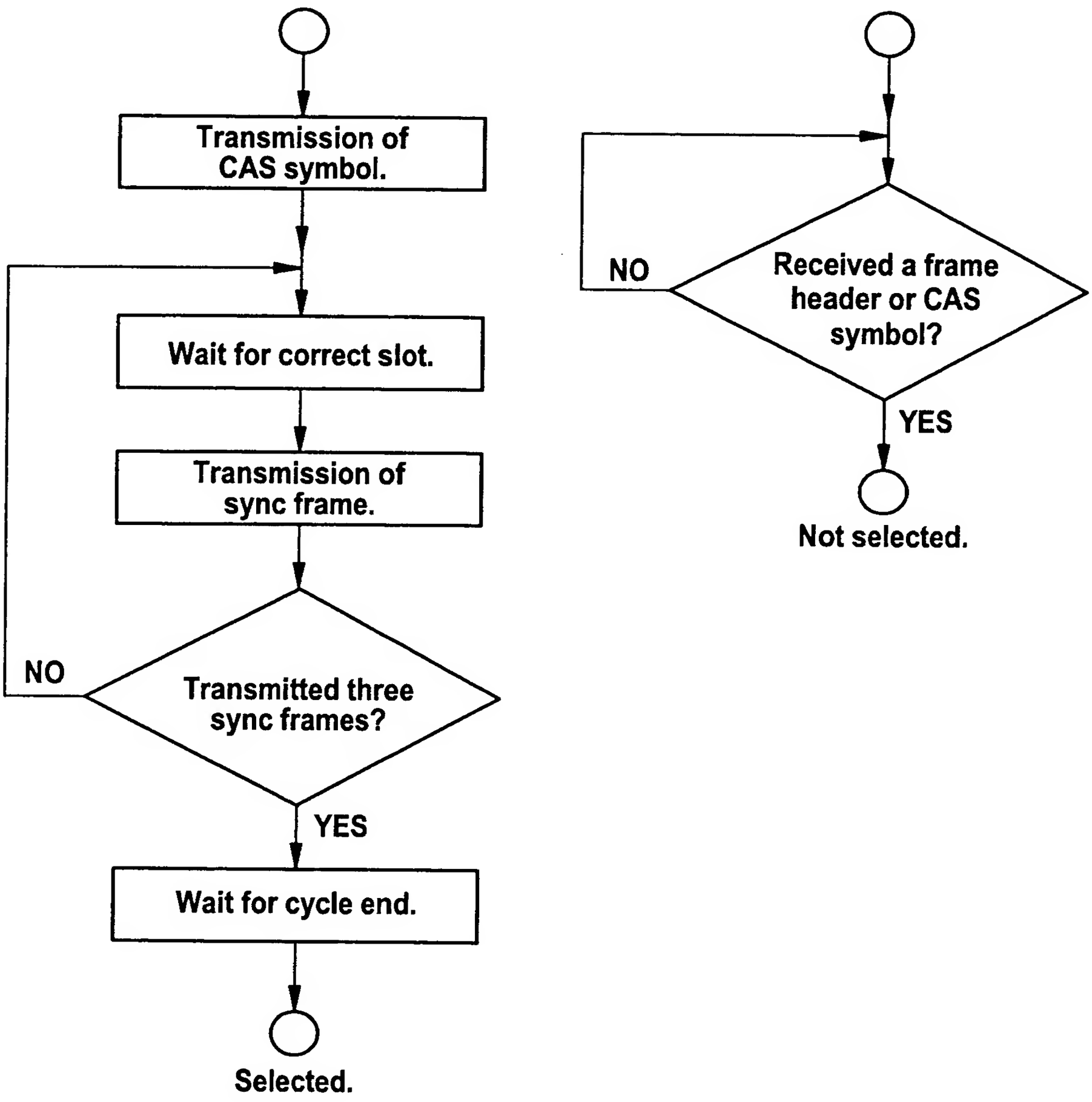


Fig. 53

Check on successfully established communication

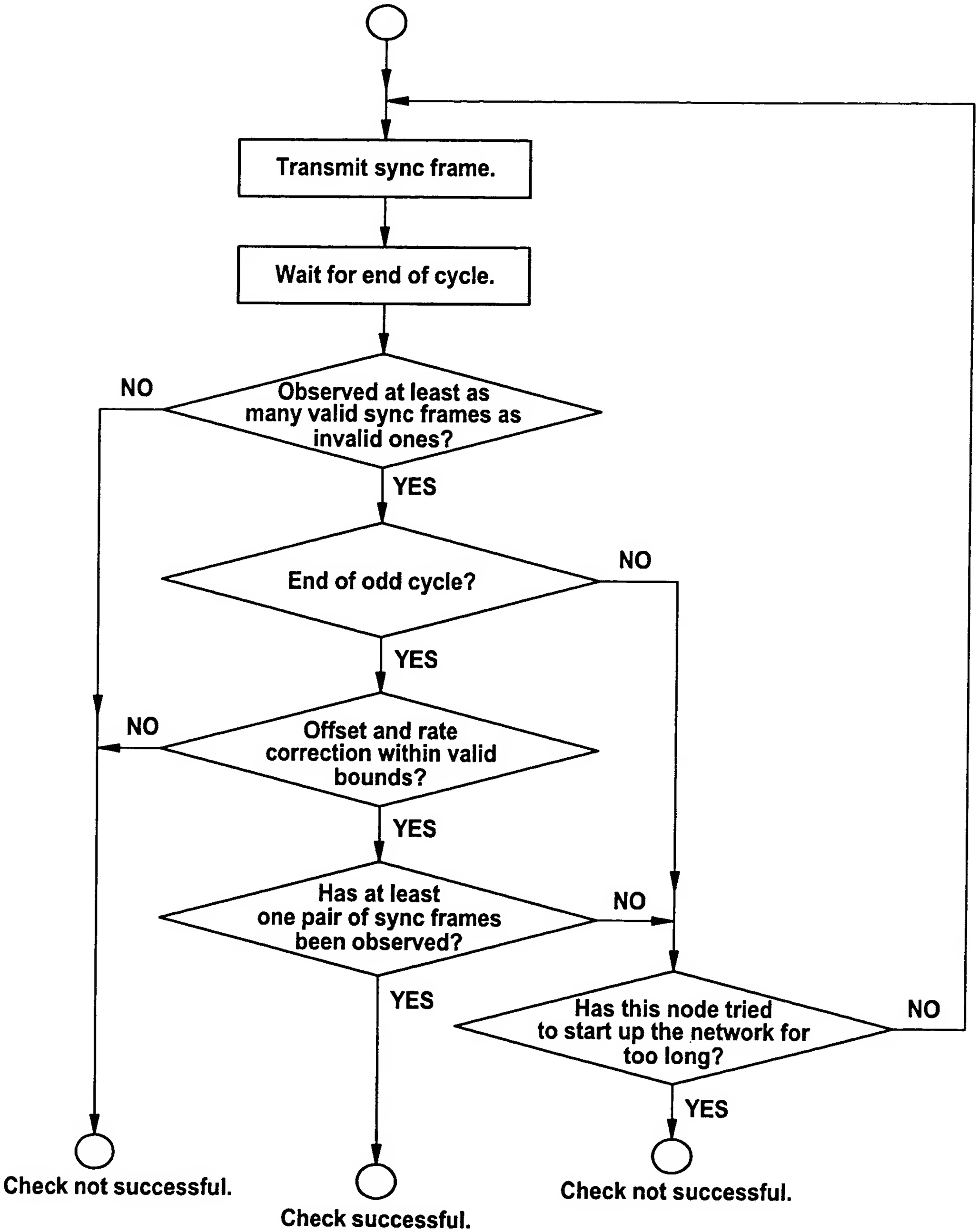


Fig. 54

Initial Synchronization

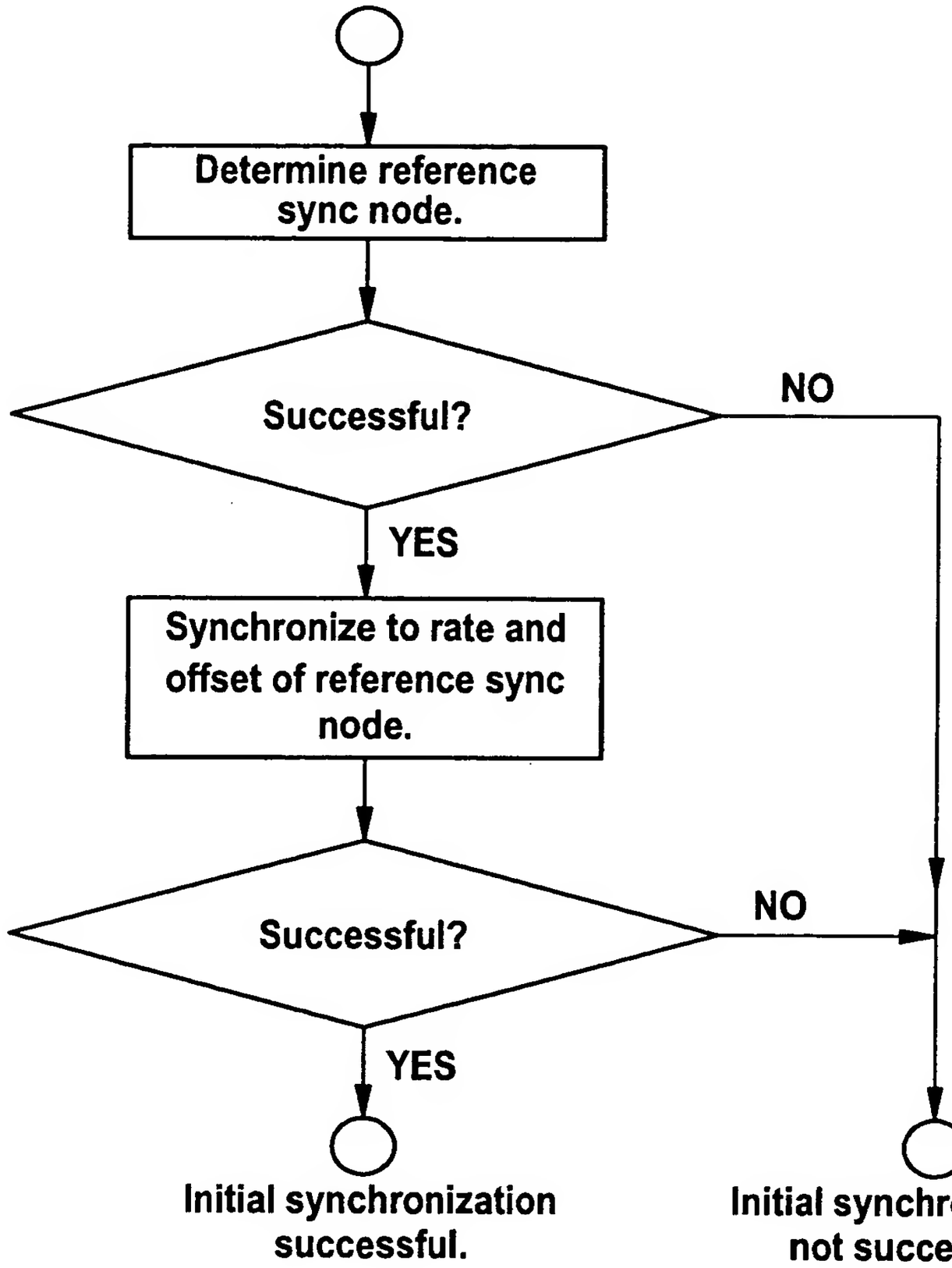
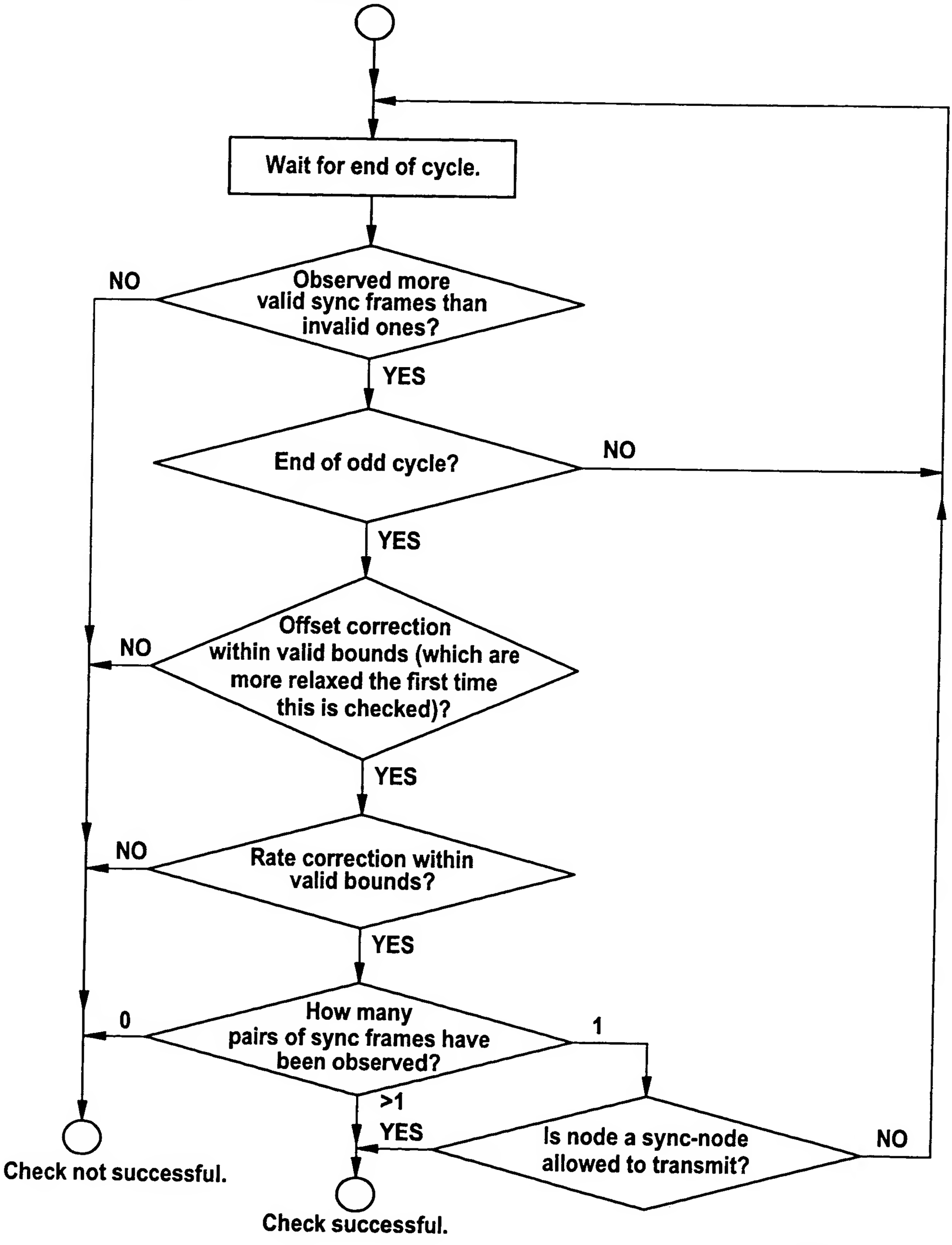


Fig. 55

Check on successful integration



06071457EP

Fig. 56

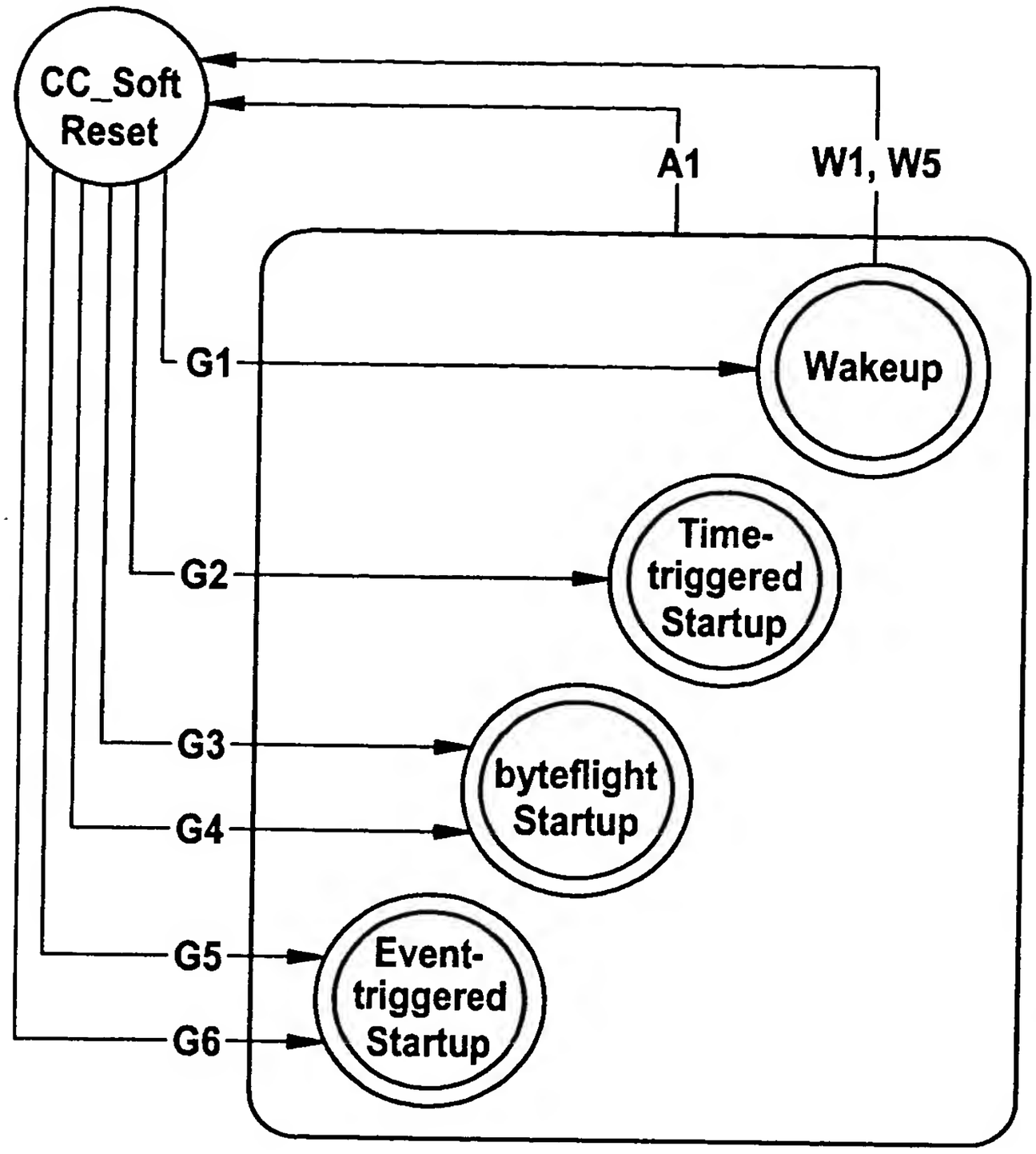

CC_Soft Reset
A1
W1, W5
G1
Wakeup
G2
Time-triggered Startup
G3
byteflight Startup
G4
G5
Event-triggered Startup
G6
Wakeup state machine for the wakeup (WU) protocol mode
Startup state machine for the time-triggered (TT-D, TT-M) protocol modes
Startup state machine for the byteflight (BF) protocol mode
Startup state machine for the event-triggered (ET) protocol mode

Fig. 57

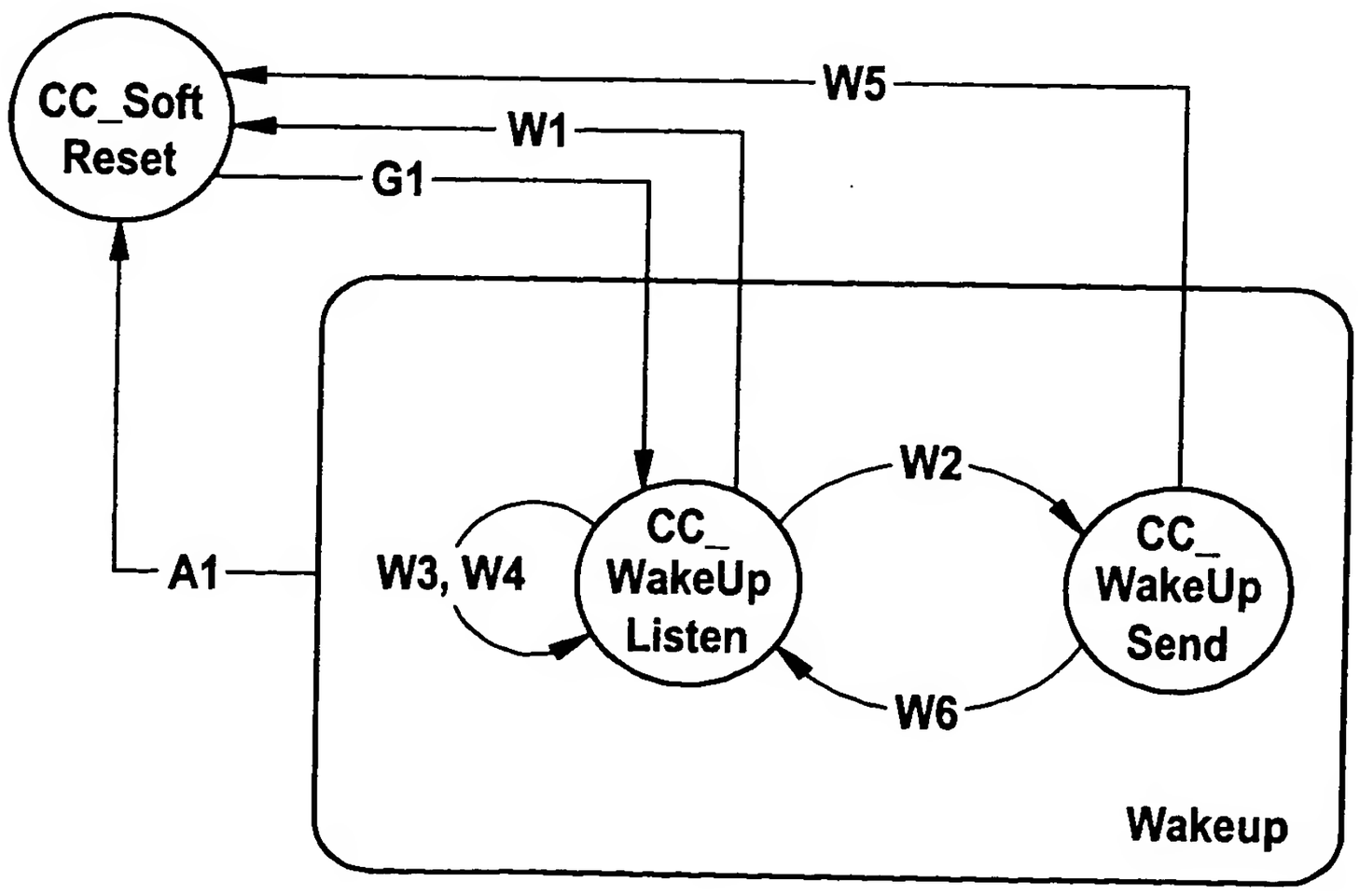
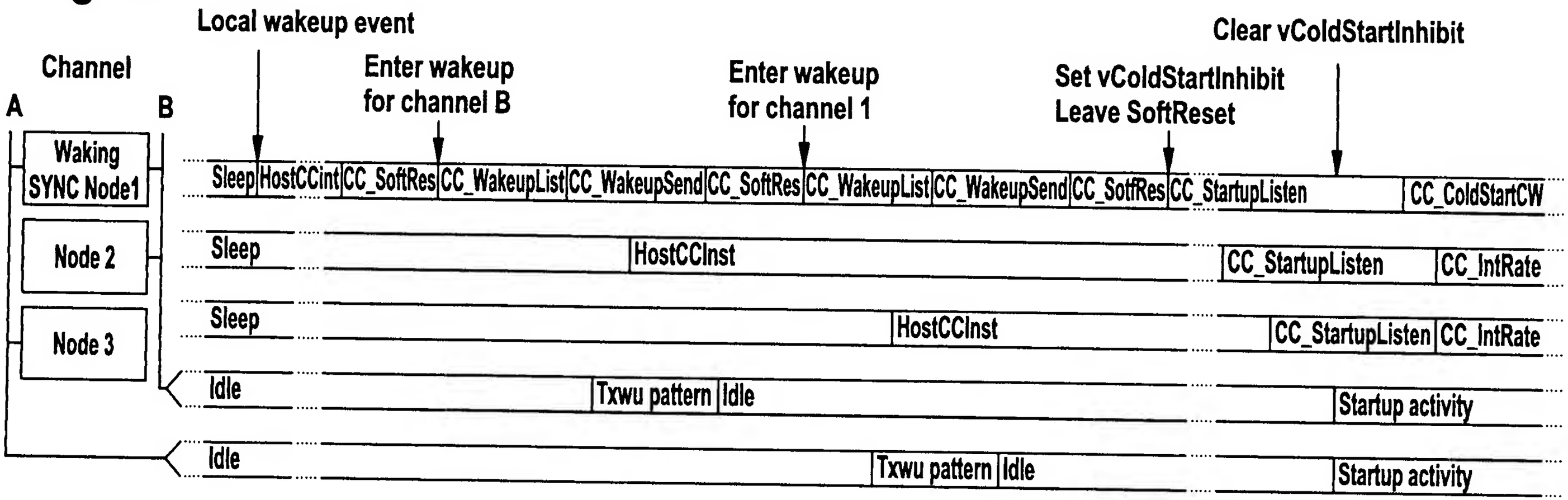
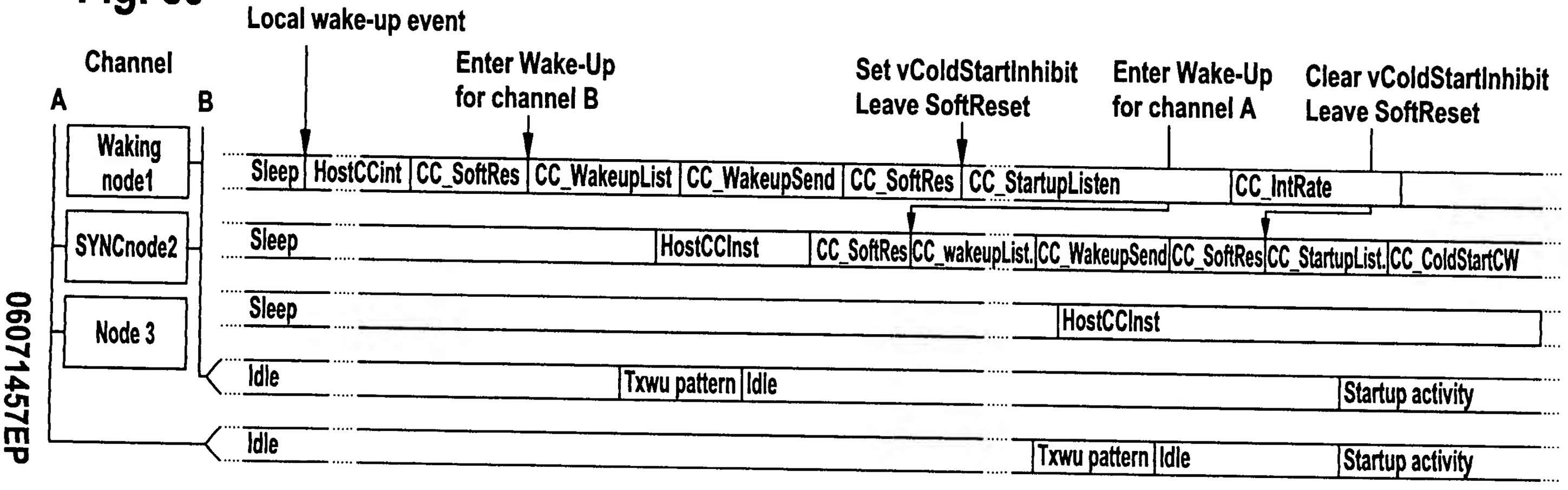

CC_Soft Reset
W5
W1
G1
A1
W3, W4
CC_ WakeUp Listen
W2
W6
CC_ WakeUp Send
Wakeup

Fig. 58



33 / 70

Fig. 59



0607145 7EP

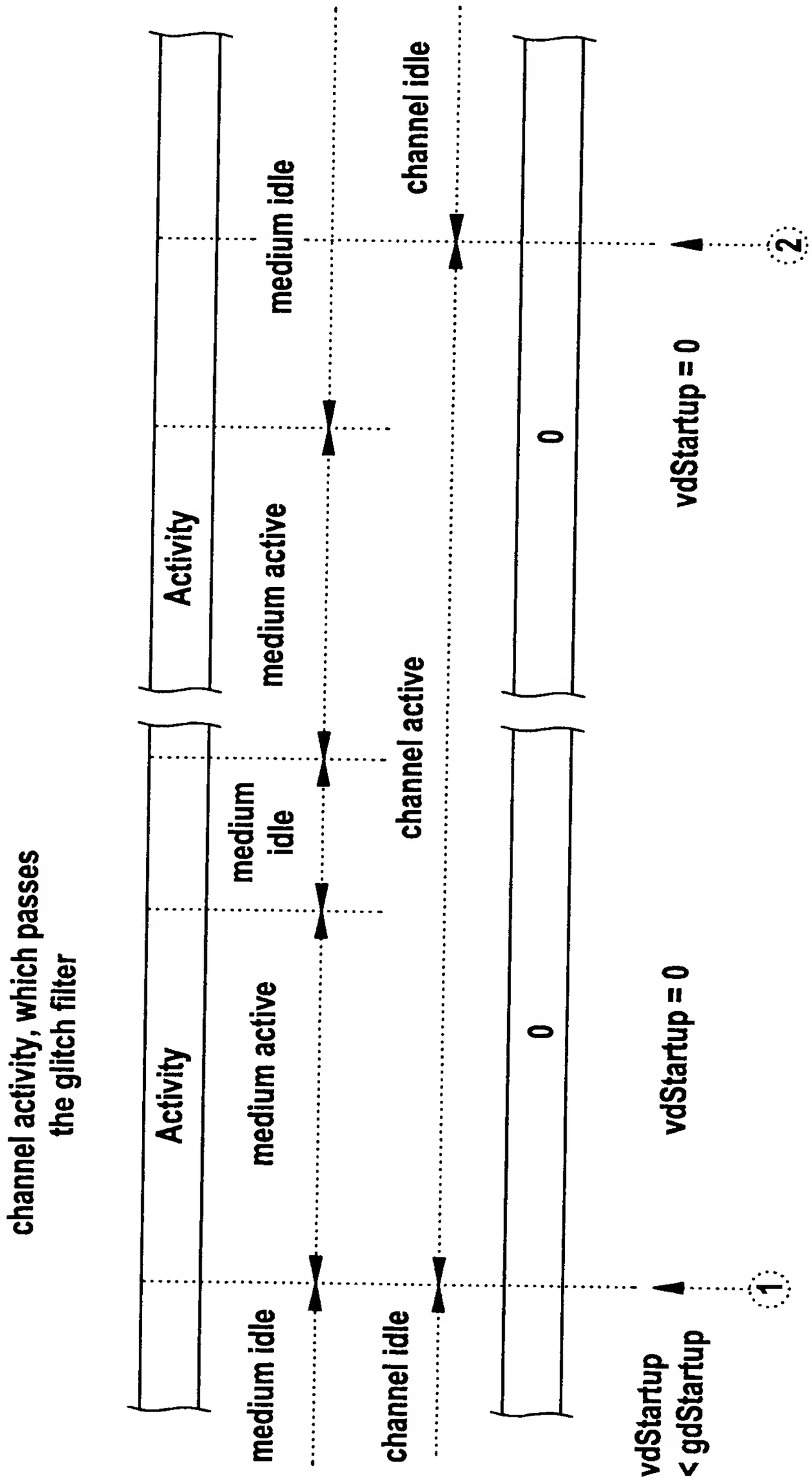

channel activity, which passes the glitch filter
Activity
medium active
medium idle
channel active
medium idle
channel idle
vdStartup < gdStartup
1
0
vdStartup = 0
2
0
vdStartup = 0

Fig. 60

Fig. 61

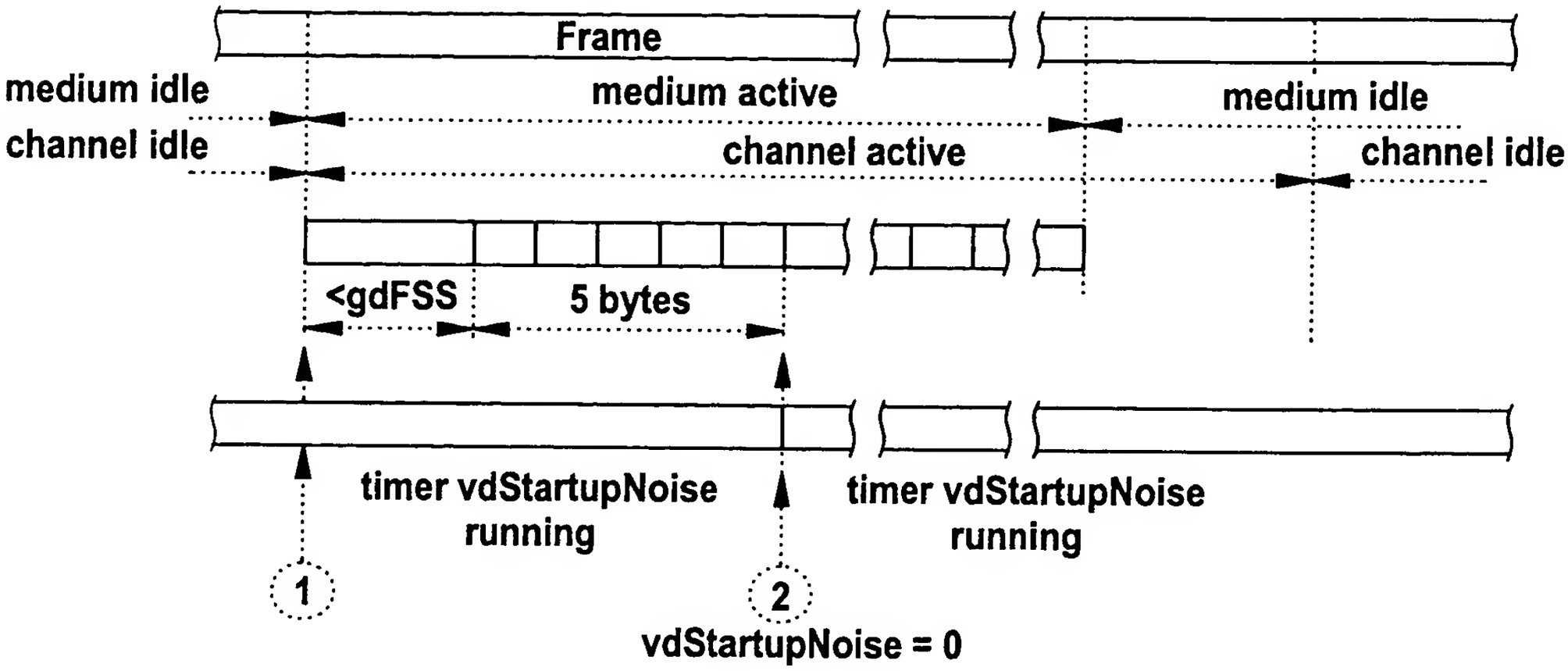

Frame
medium idle
medium active
medium idle
channel idle
channel active
channel idle
<gdFSS
5 bytes
timer vdStartupNoise running
timer vdStartupNoise running
1
2
vdStartupNoise = 0

Fig. 62

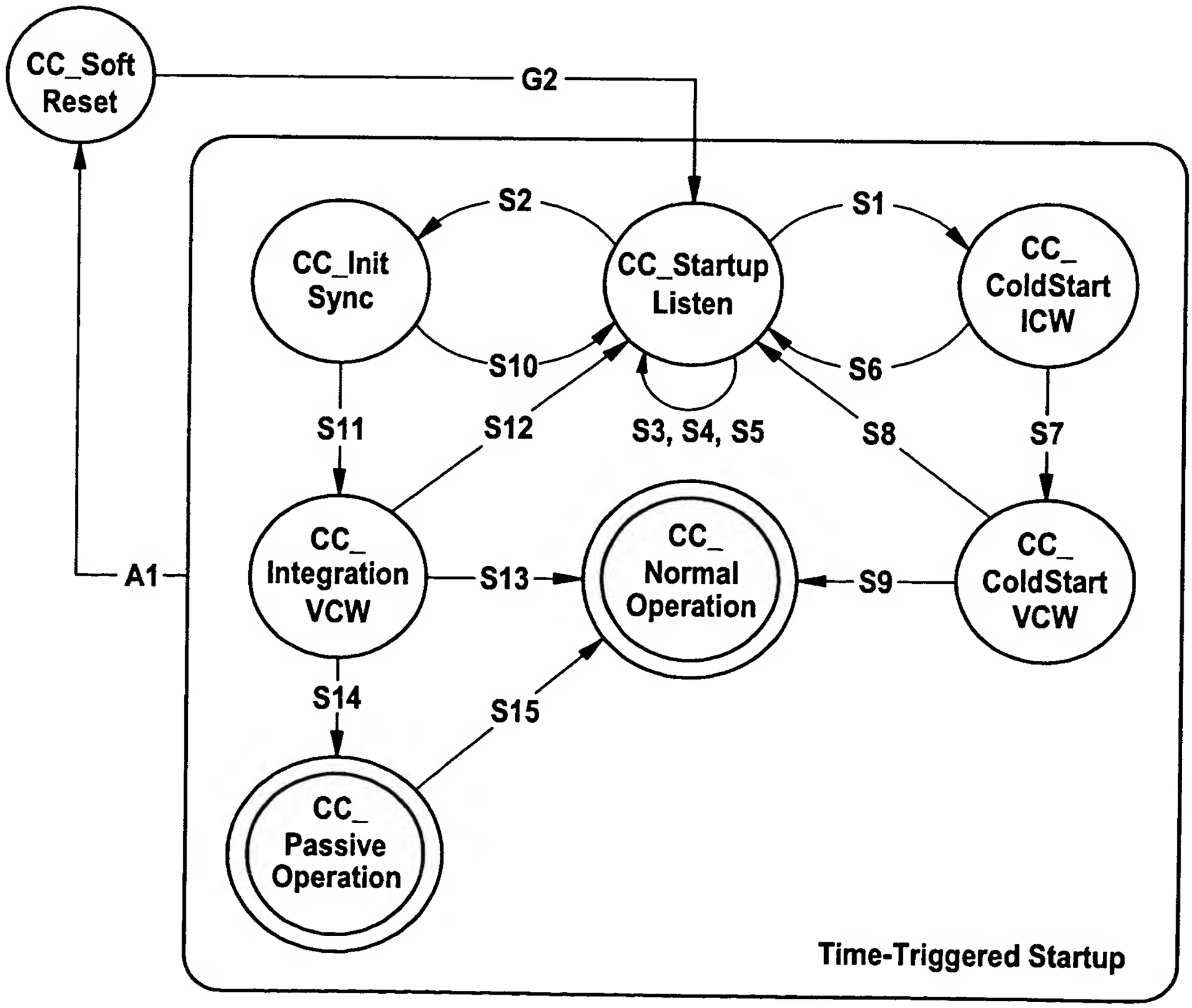

CC_Soft Reset
G2
S2
S1
CC_Init Sync
CC_Startup Listen
CC_ ColdStart ICW
S10
S6
S11
S12
S3, S4, S5
S8
S7
A1
CC_ Integration VCW
S13
CC_ Normal Operation
S9
CC_ ColdStart VCW
S14
S15
CC_ Passive Operation
Time-Triggered Startup

Fig. 63

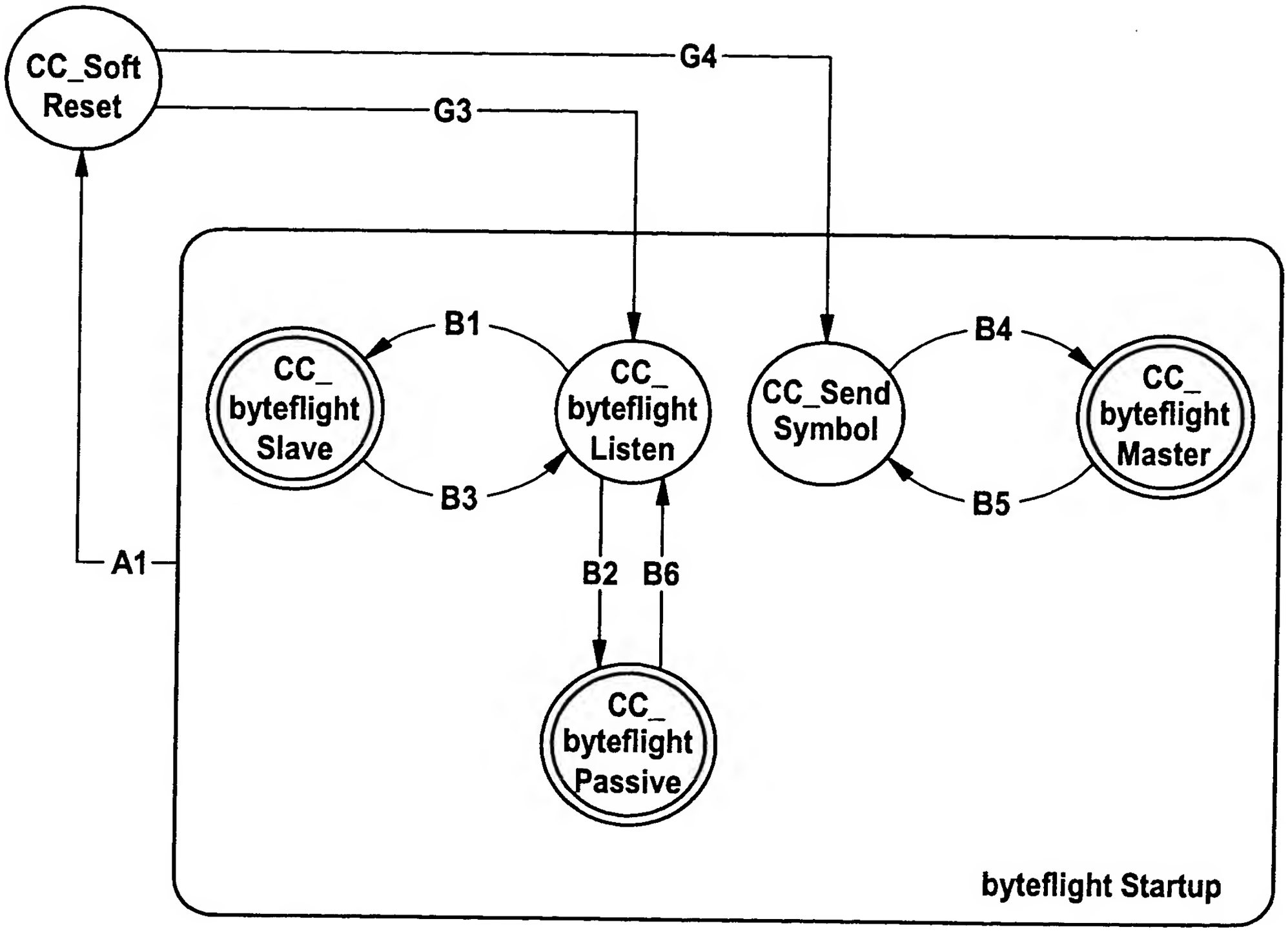
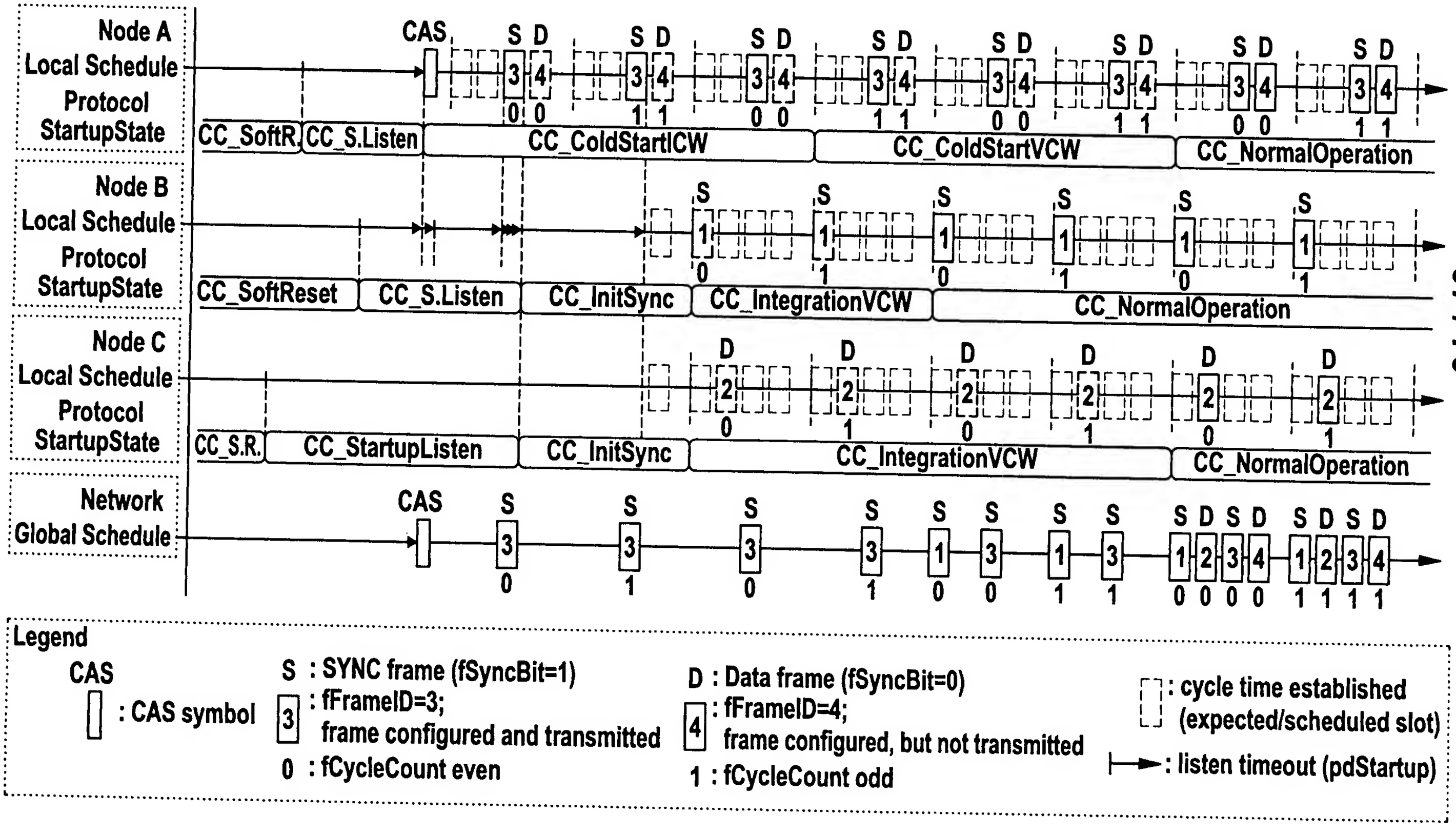
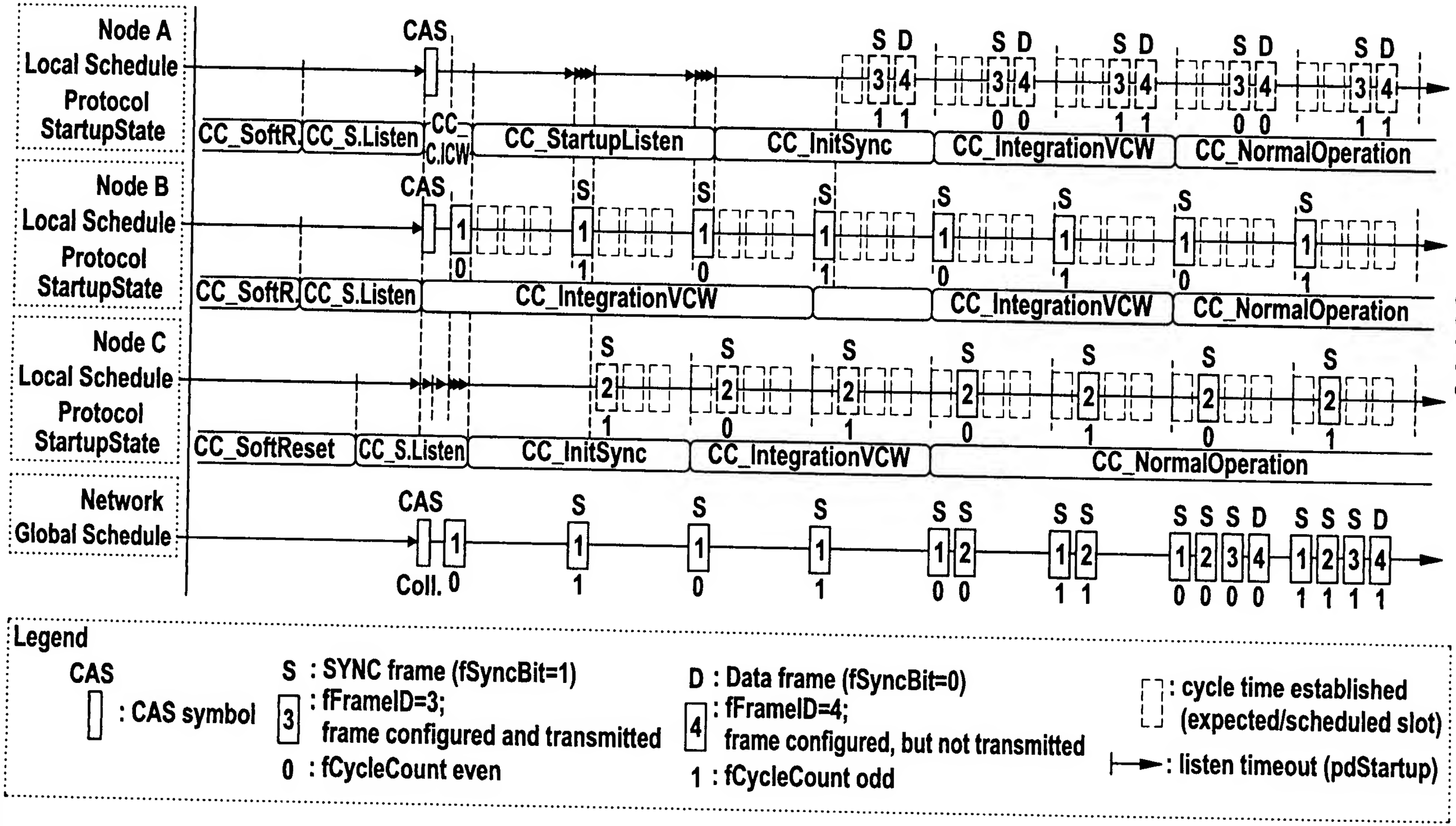

CC_Soft Reset
G4
G3
CC_ byteflight Slave
B1
B3
CC_ byteflight Listen
CC_Send Symbol
B4
B5
CC_ byteflight Master
A1
B2
B6
CC_ byteflight Passive
byteflight Startup

Fig. 64



0607145/EP

Fig. 65



Legend

CAS : CAS symbol

S : SYNC frame (fSyncBit=1)

3 : fFrameID=3; frame configured and transmitted

0 : fCycleCount even

D : Data frame (fSyncBit=0)

4 : fFrameID=4; frame configured, but not transmitted

1 : fCycleCount odd

: cycle time established (expected/scheduled slot)

: listen timeout (pdStartup)

Fig. 66

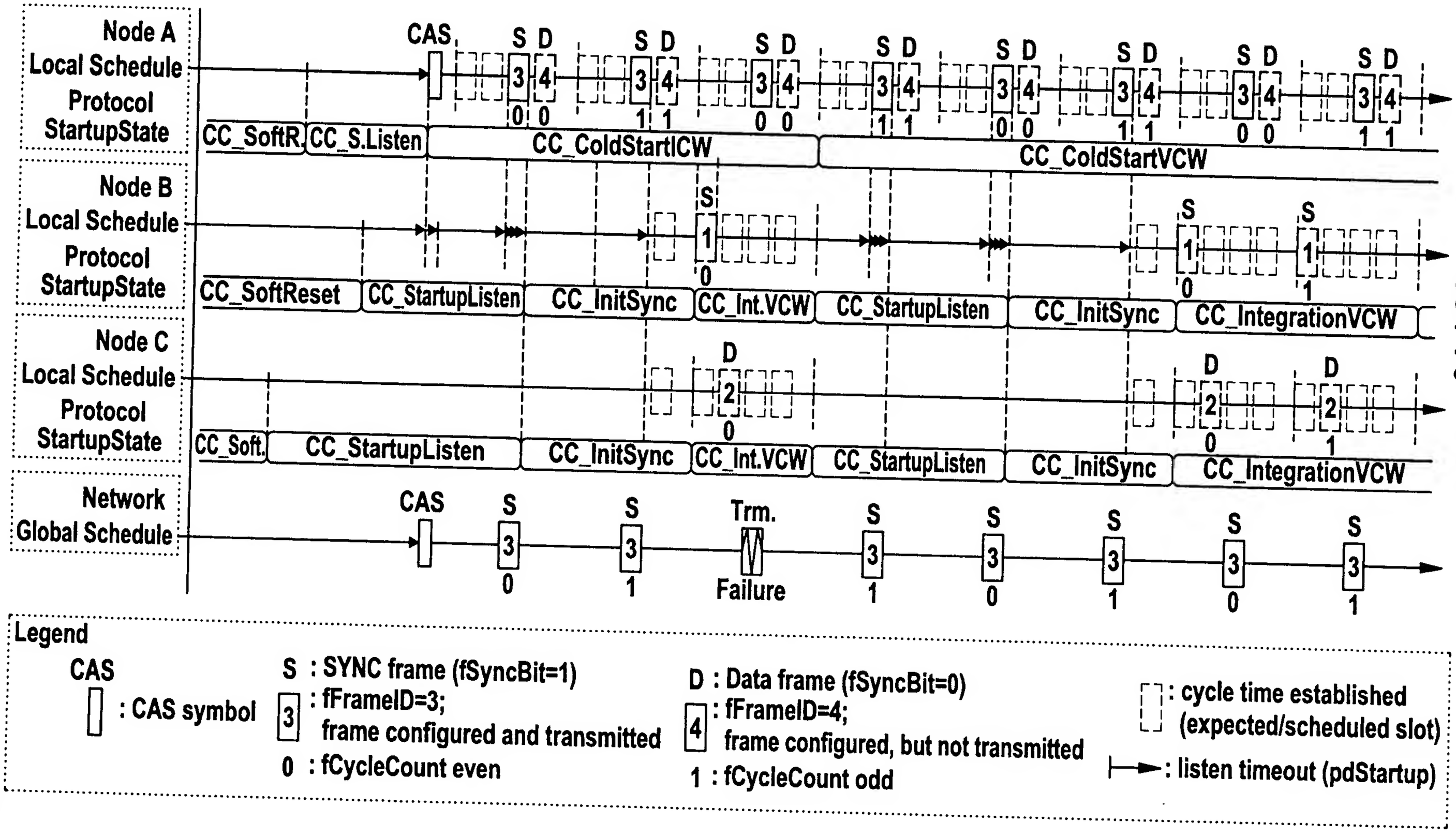

Node A
Local Schedule
Protocol StartupState
CAS
S D
CC_SoftR
CC_S.Listen
CC_ColdStartICW
CC_ColdStartVCW
Node B
Local Schedule
Protocol StartupState
S
CC_SoftReset
CC_StartupListen
CC_InitSync
CC_Int.VCW
CC_StartupListen
CC_InitSync
CC_IntegrationVCW
Node C
Local Schedule
Protocol StartupState
D
CC_Soft.
CC_StartupListen
CC_InitSync
CC_Int.VCW
CC_StartupListen
CC_InitSync
CC_IntegrationVCW
Network
Global Schedule
CAS
S
Trm.
Failure
Legend
CAS
: CAS symbol
S : SYNC frame (fSyncBit=1)
: fFrameID=3;
frame configured and transmitted
0 : fCycleCount even
D : Data frame (fSyncBit=0)
: fFrameID=4;
frame configured, but not transmitted
1 : fCycleCount odd
: cycle time established
(expected/scheduled slot)
: listen timeout (pdStartup)

Fig. 67

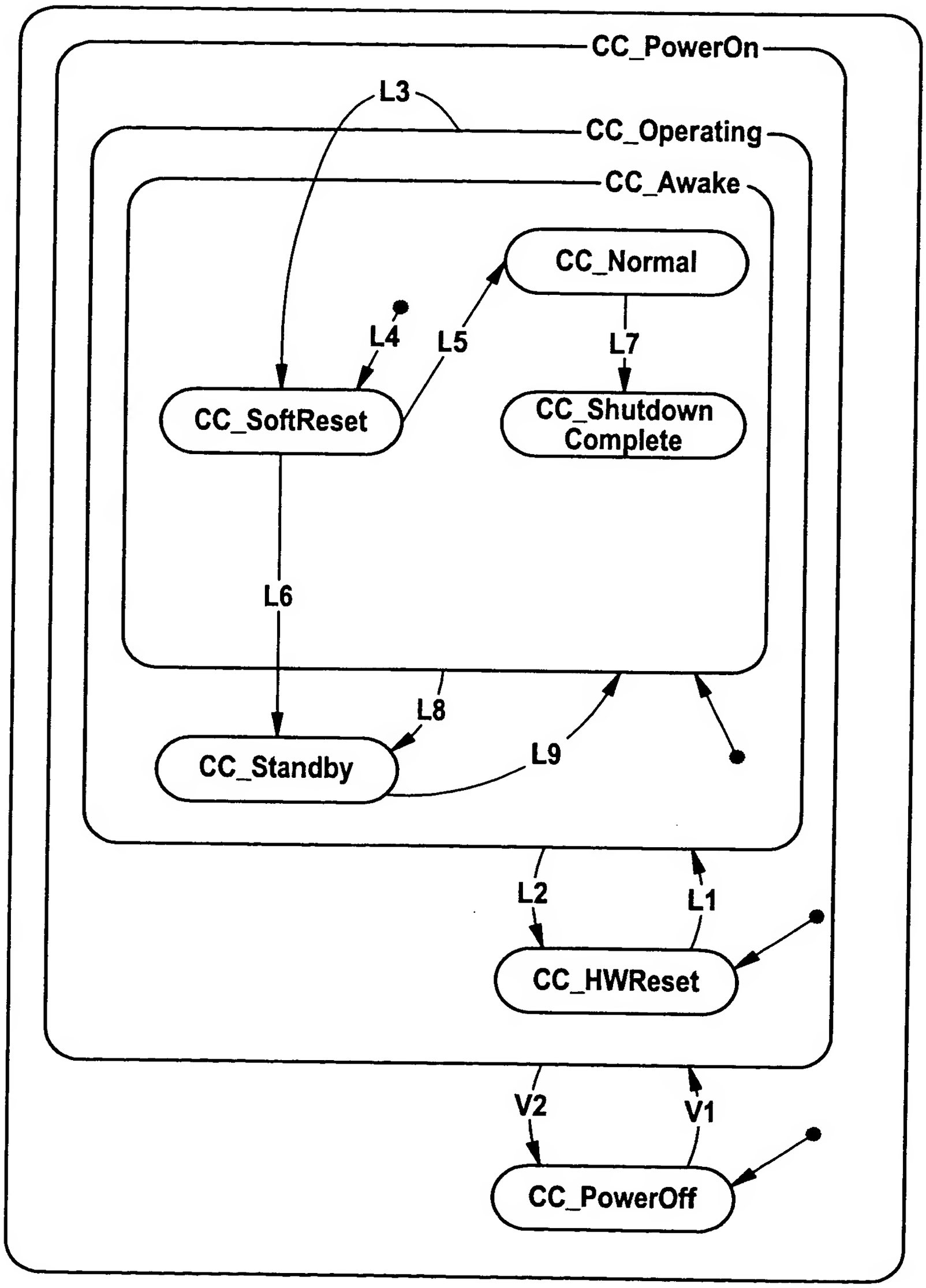

CC_PowerOn
L3
CC_Operating
CC_Awake
CC_Normal
L4
L5
L7
CC_SoftReset
CC_Shutdown Complete
L6
L8
CC_Standby
L9
L2
L1
CC_HWReset
V2
V1
CC_PowerOff

Fig. 68

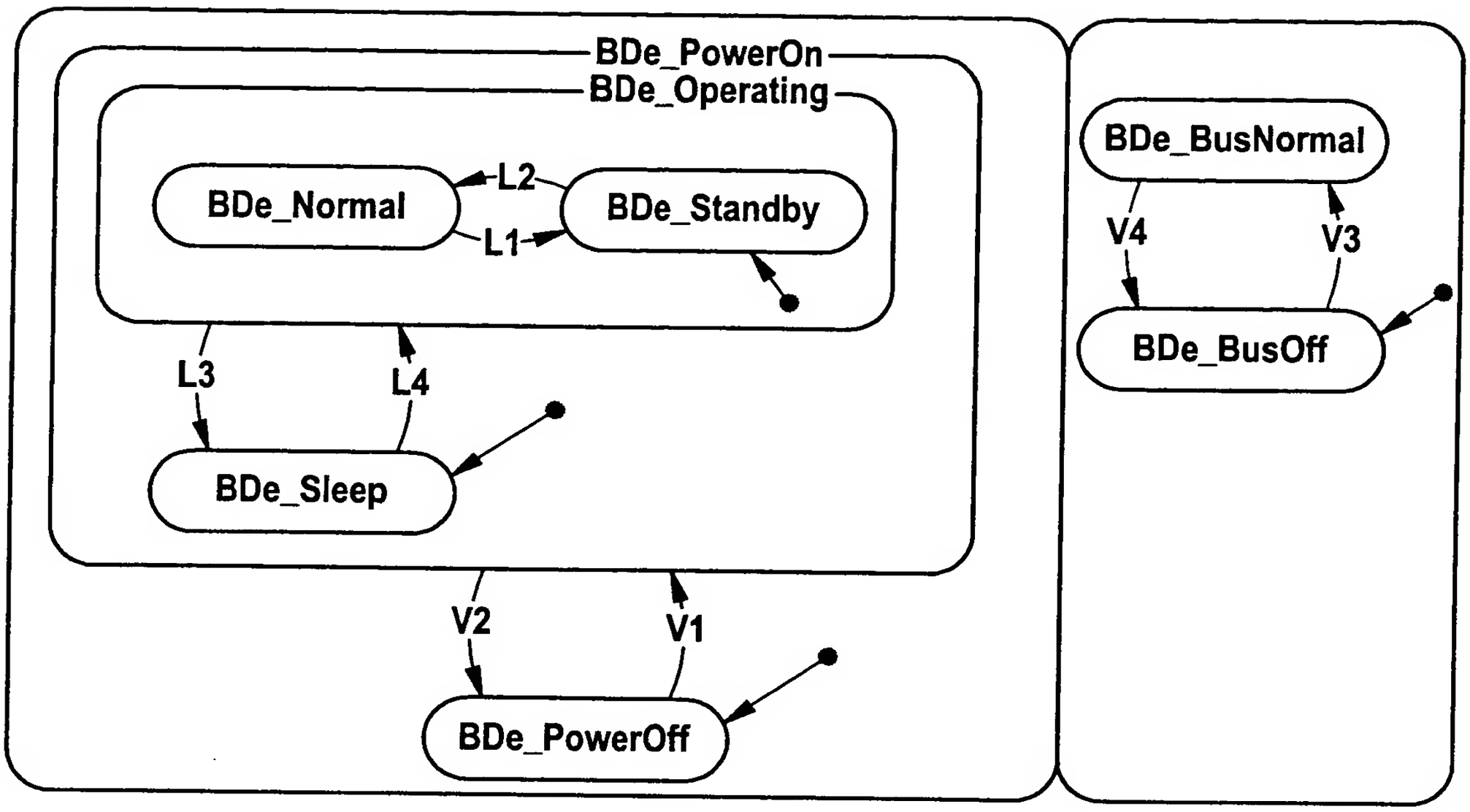

BDe_PowerOn
BDe_Operating
BDe_Normal
L2
L1
BDe_Standby
L3
L4
BDe_Sleep
V2
V1
BDe_PowerOff
BDe_BusNormal
V4
V3
BDe_BusOff

Fig. 69

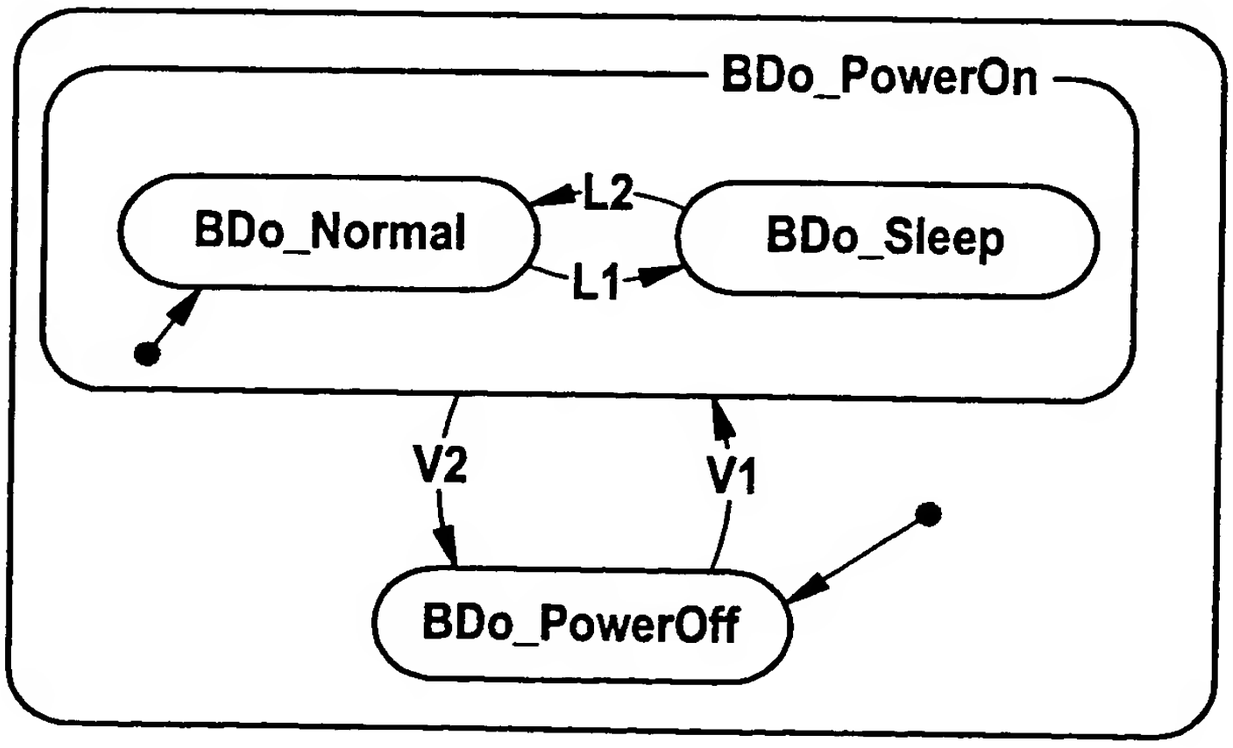

BDo_PowerOn
BDo_Normal
L2
L1
BDo_Sleep
V2
V1
BDo_PowerOff

Fig. 70

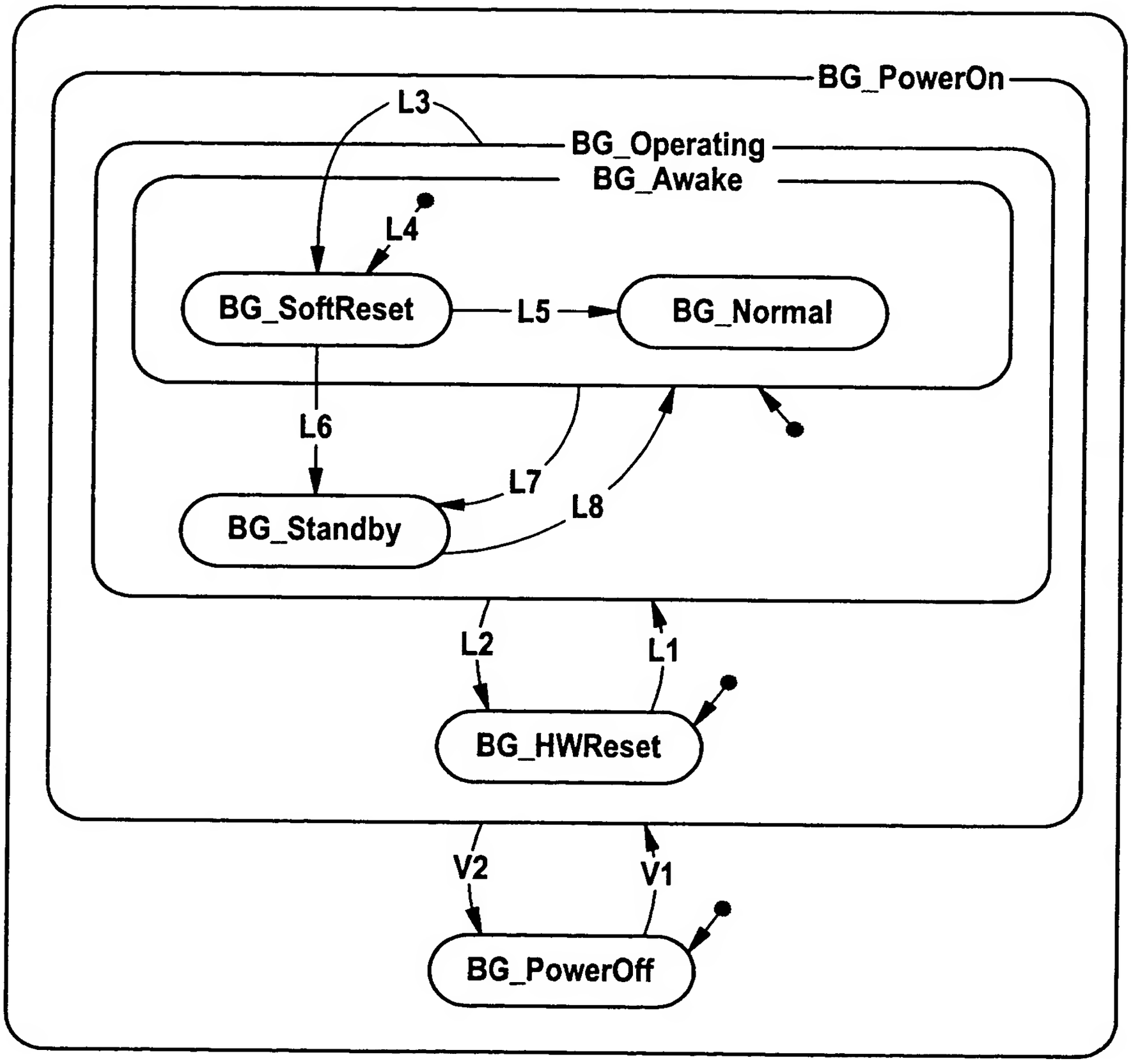

BG_PowerOn
L3
BG_Operating
BG_Awake
L4
BG_SoftReset
L5
BG_Normal
L6
L7
L8
BG_Standby
L2
L1
BG_HWReset
V2
V1
BG_PowerOff

Fig. 71

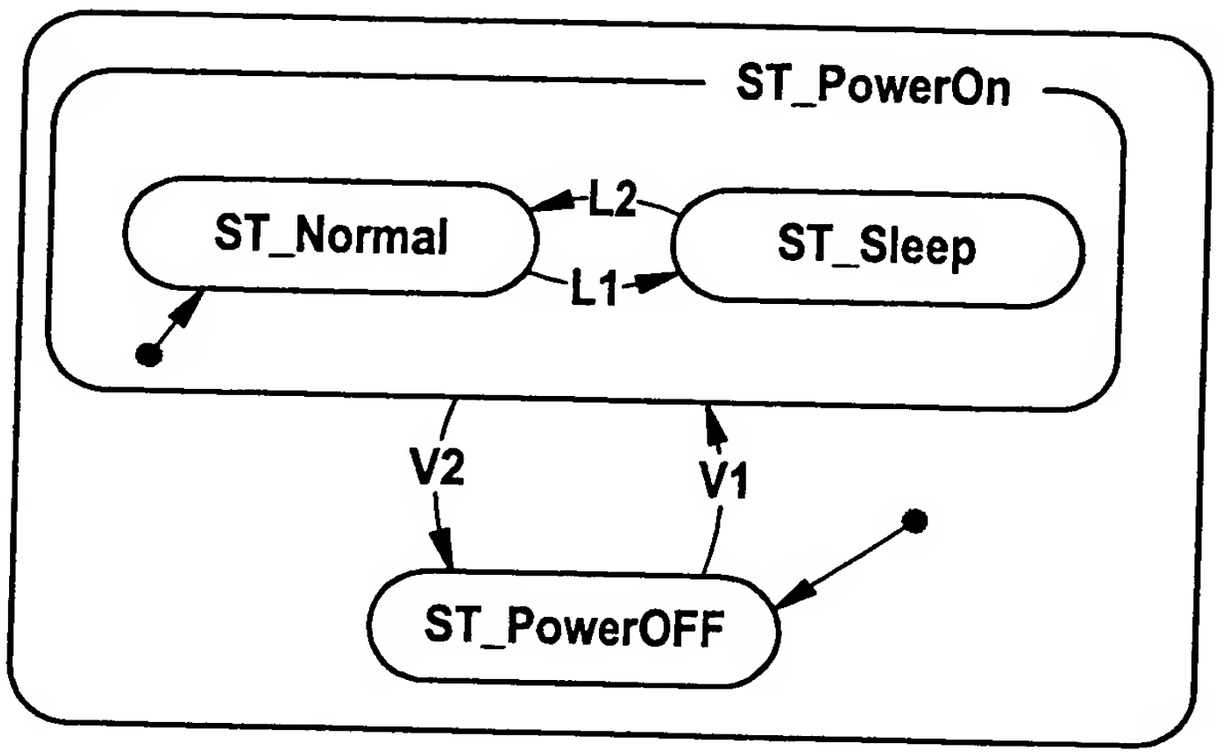


Fig. 72

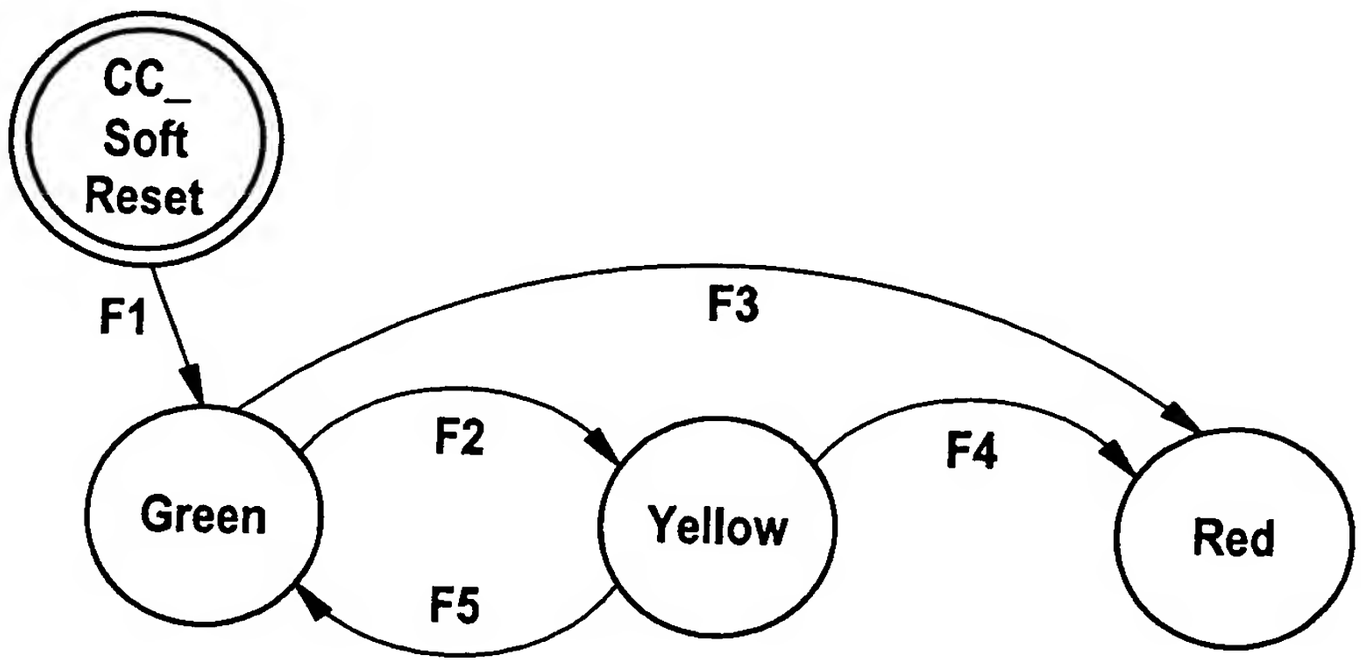


Fig. 73

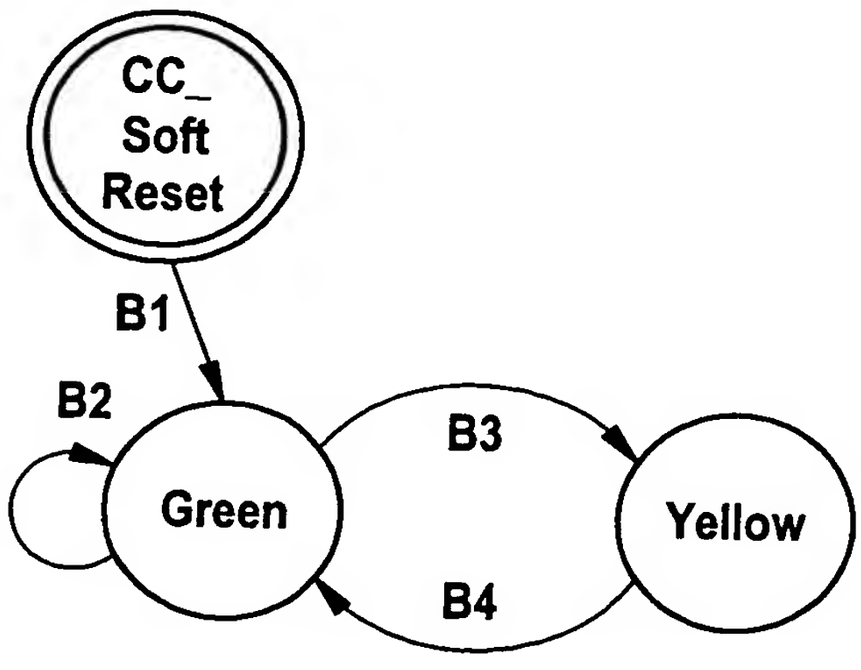


Fig. 74

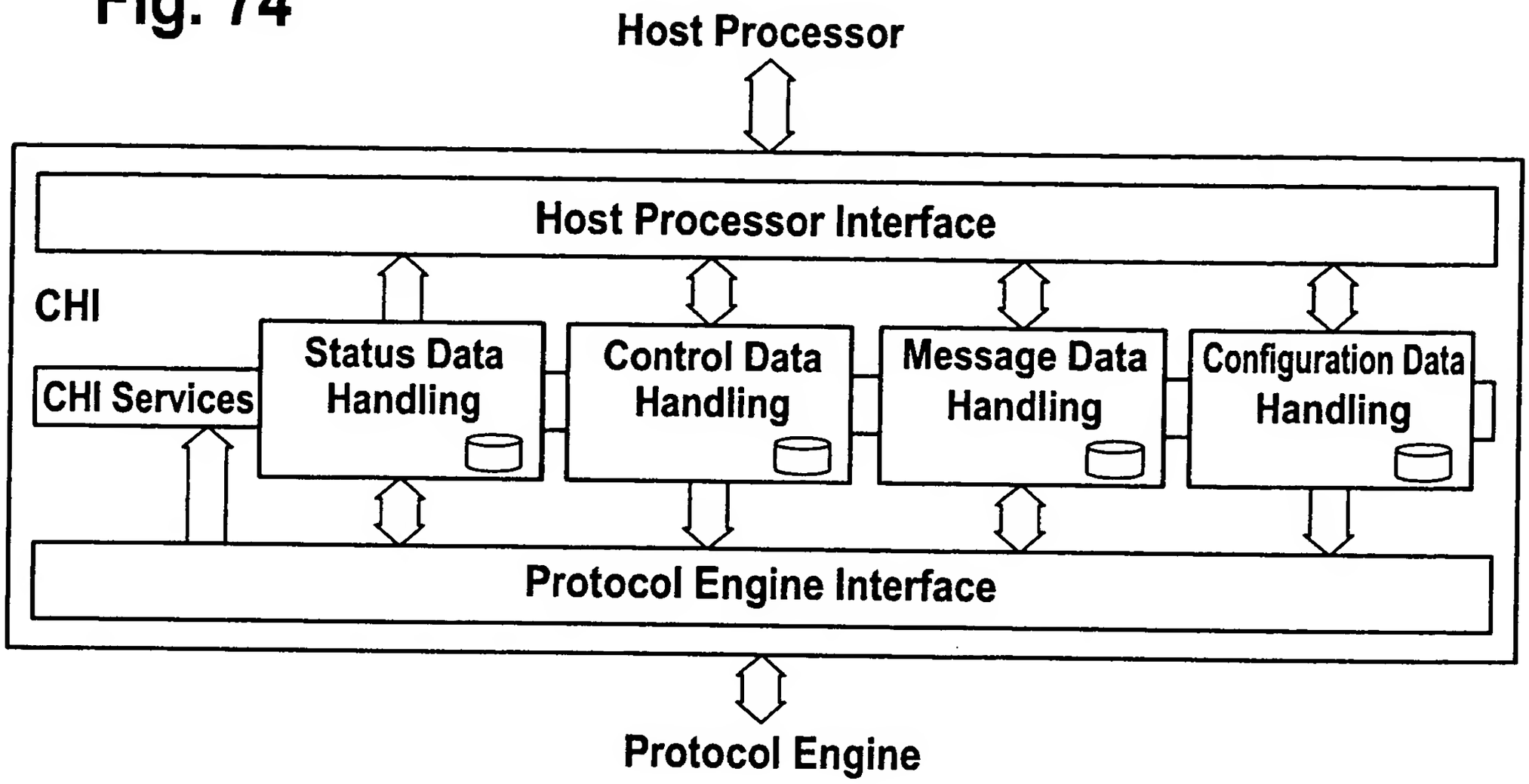

Host Processor
Host Processor Interface
CHI
CHI Services
Status Data Handling
Control Data Handling
Message Data Handling
Configuration Data Handling
Protocol Engine Interface
Protocol Engine

Fig. 75

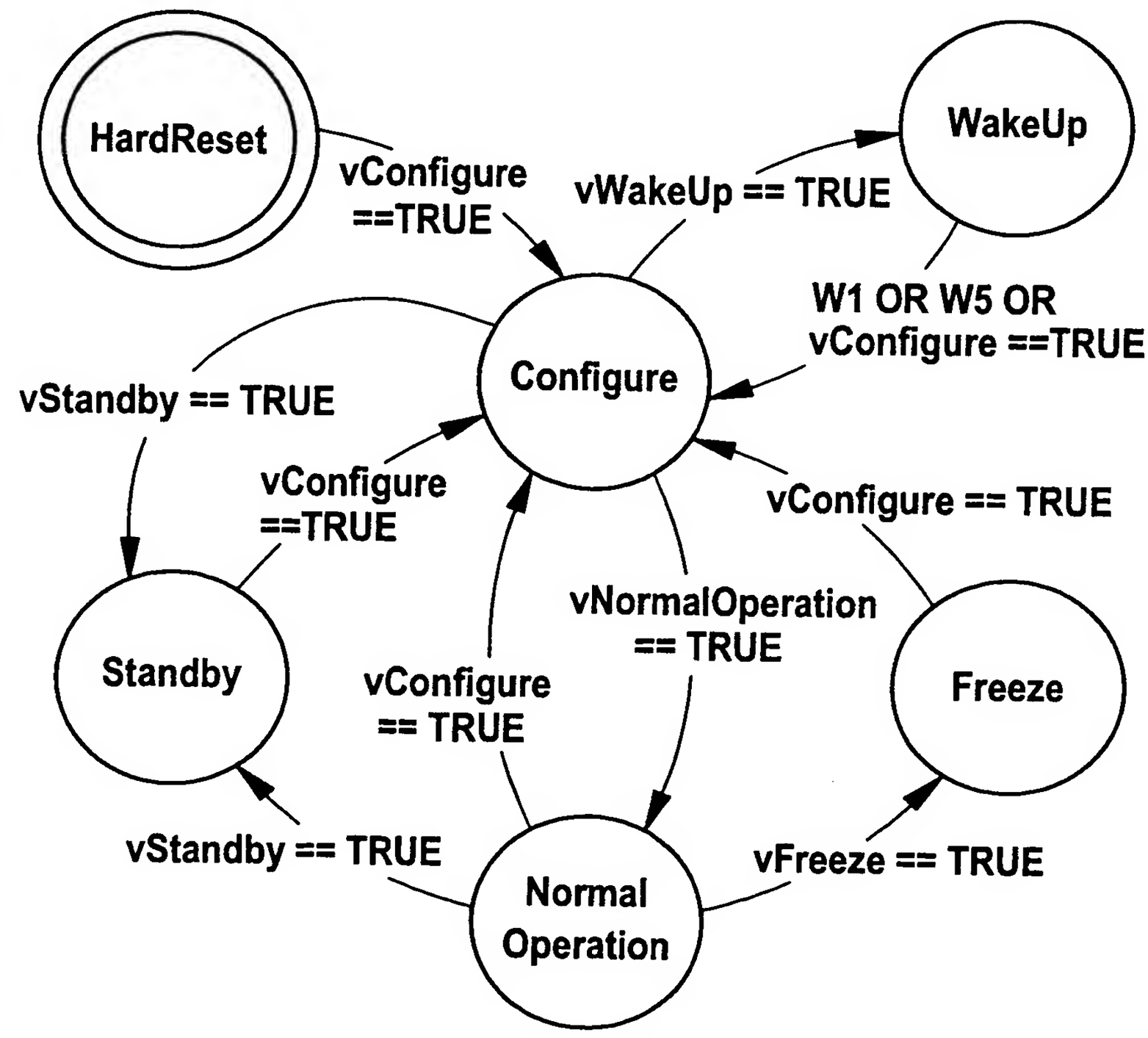

HardReset
vConfigure ==TRUE
WakeUp
vWakeUp == TRUE
W1 OR W5 OR vConfigure ==TRUE
Configure
vStandby == TRUE
vConfigure ==TRUE
vConfigure == TRUE
vNormalOperation == TRUE
Standby
vConfigure == TRUE
Freeze
vStandby == TRUE
vFreeze == TRUE
Normal Operation

Fig. 76

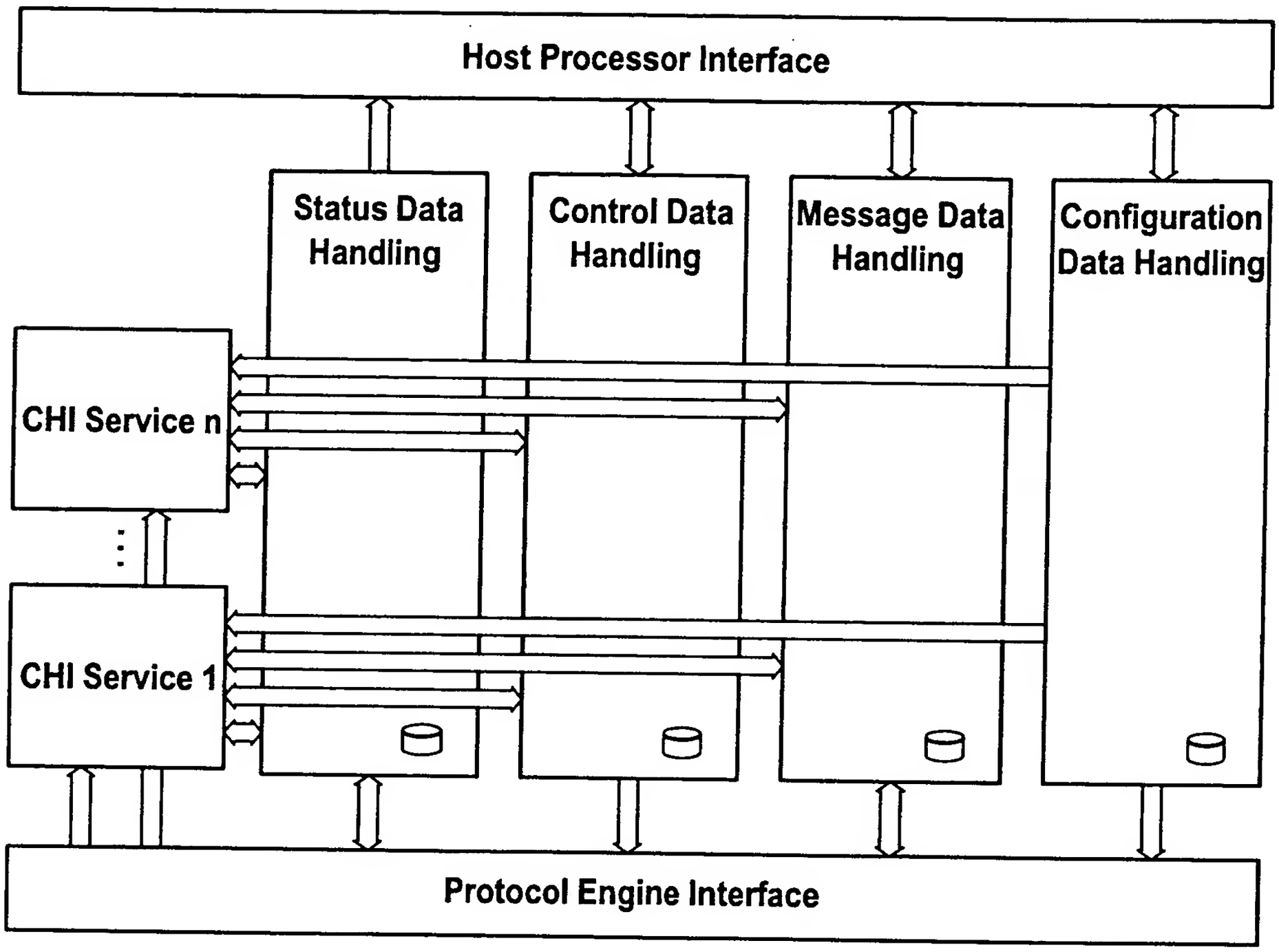

Host Processor Interface
Status Data Handling
Control Data Handling
Message Data Handling
Configuration Data Handling
CHI Service n
CHI Service 1
Protocol Engine Interface

Fig. 77

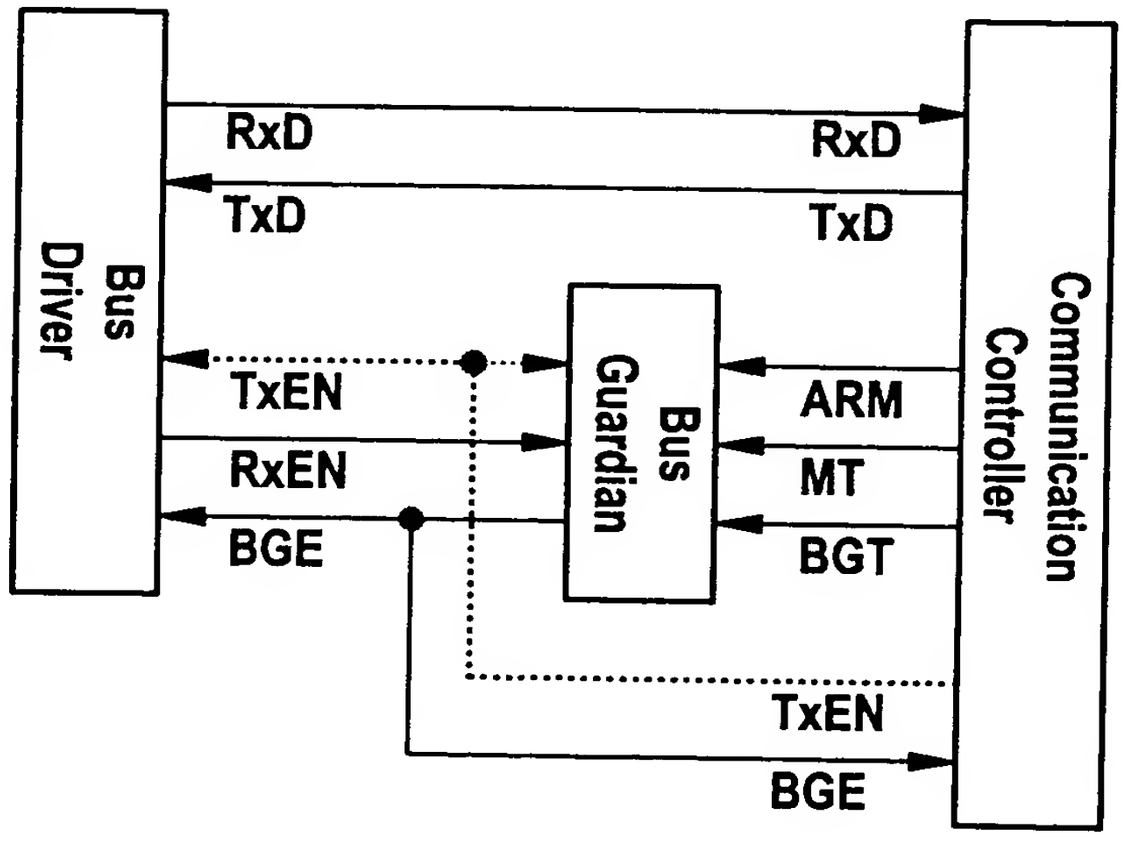

Bus Driver
RxD
TxD
TxEN
RxEN
BGE
Bus Guardian
RxD
TxD
ARM
MT
BGT
TxEN
BGE
Communication Controller

Fig. 78

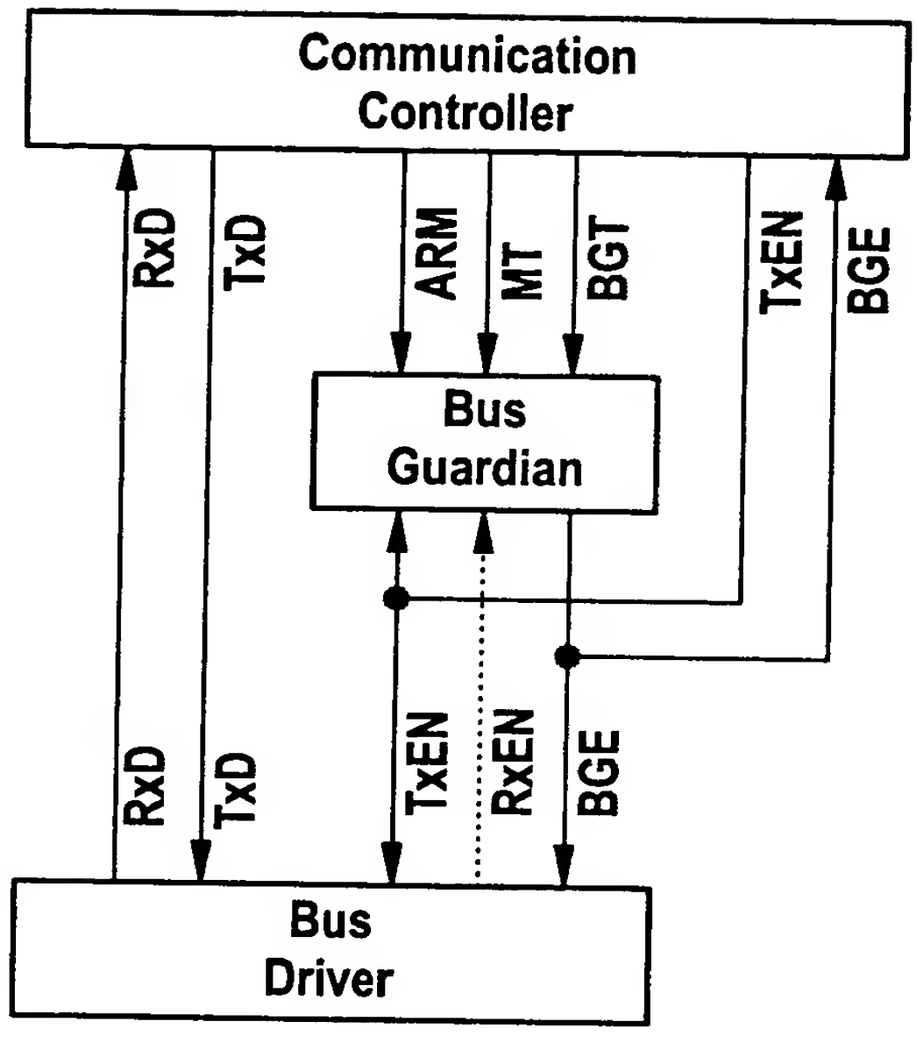


Fig. 79

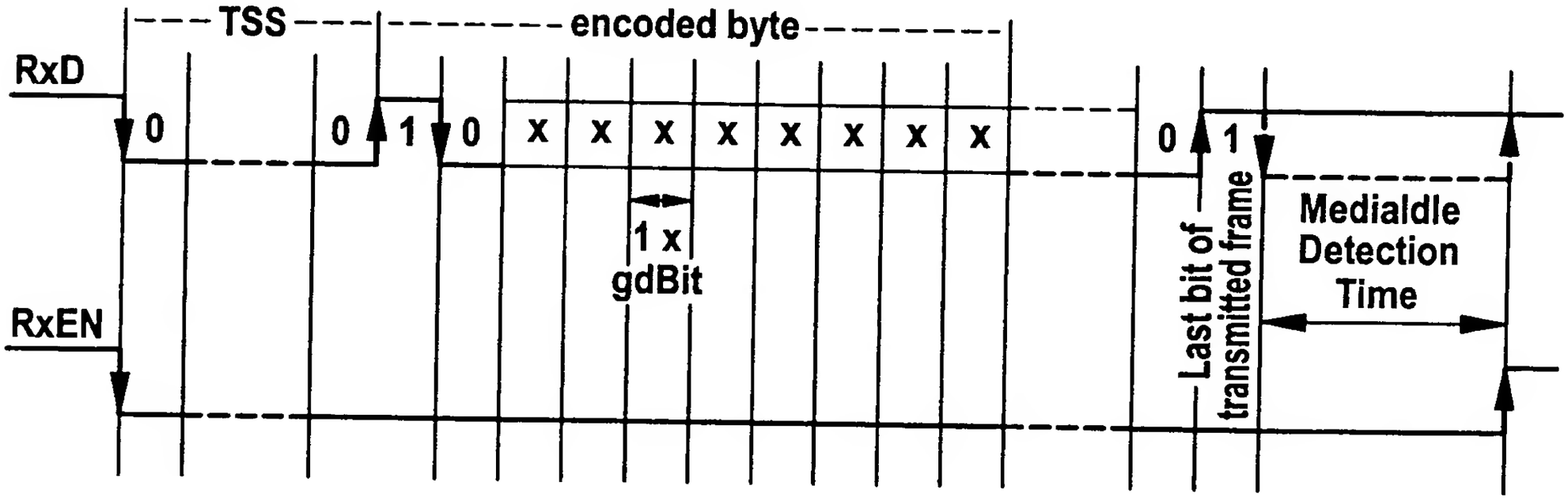


Fig. 80

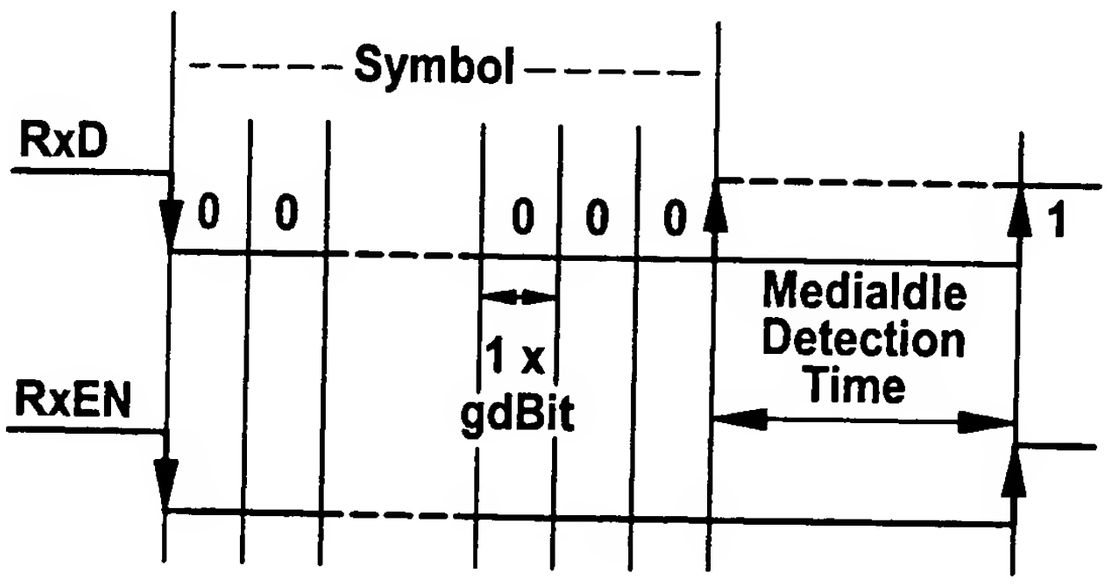


Fig. 81

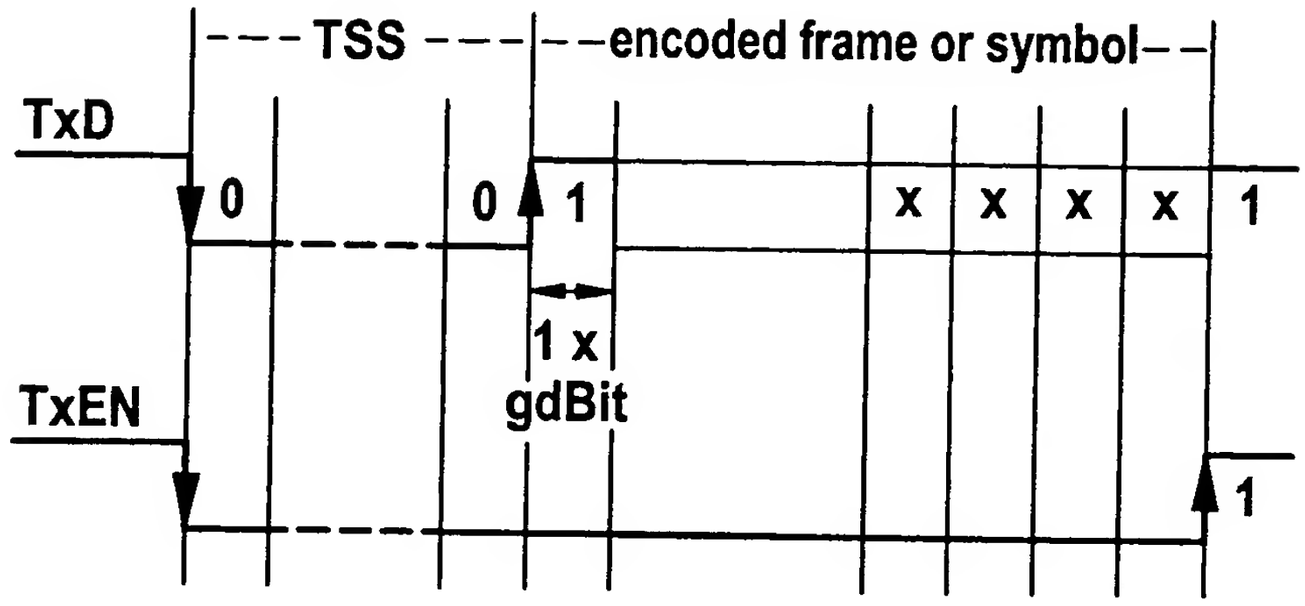
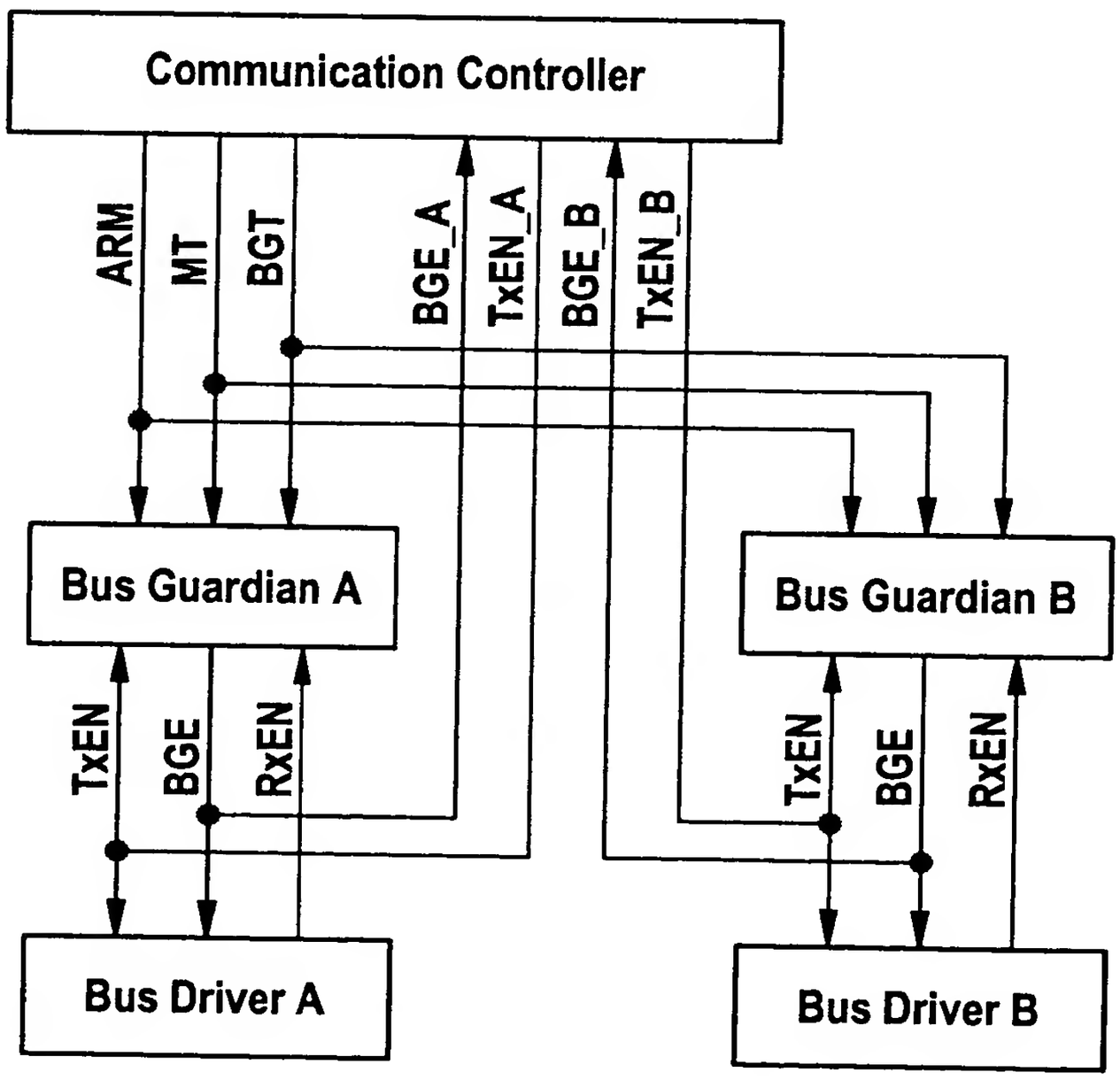

TSS
encoded frame or symbol
TxD
0
0
1
x
x
x
x
1
1 x
gdBit
TxEN
1

Fig. 82


Communication Controller
ARM
MT
BGT
BGE_A
TxEN_A
BGE_B
TxEN_B
Bus Guardian A
Bus Guardian B
TxEN
BGE
RxEN
TxEN
BGE
RxEN
Bus Driver A
Bus Driver B

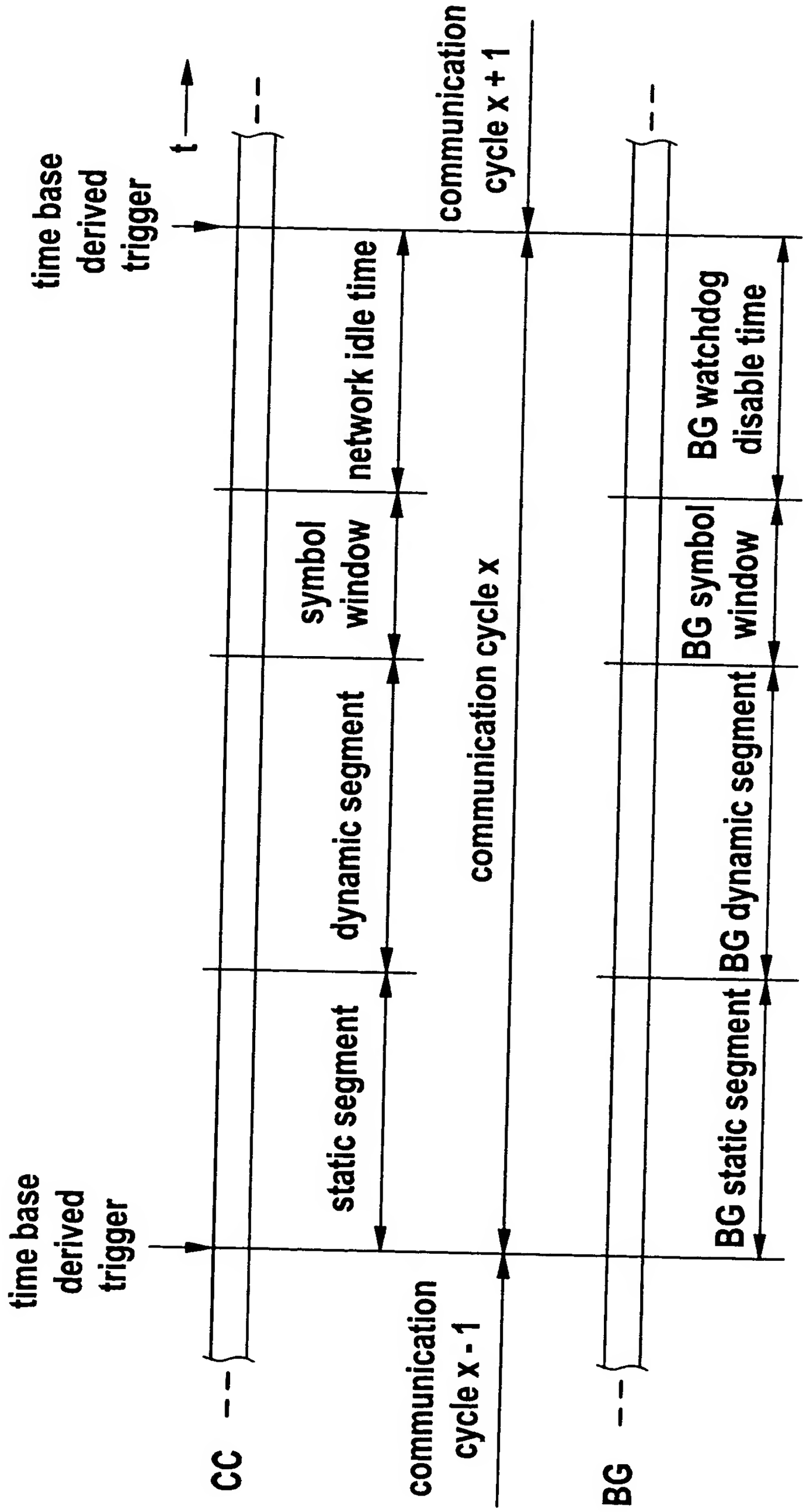
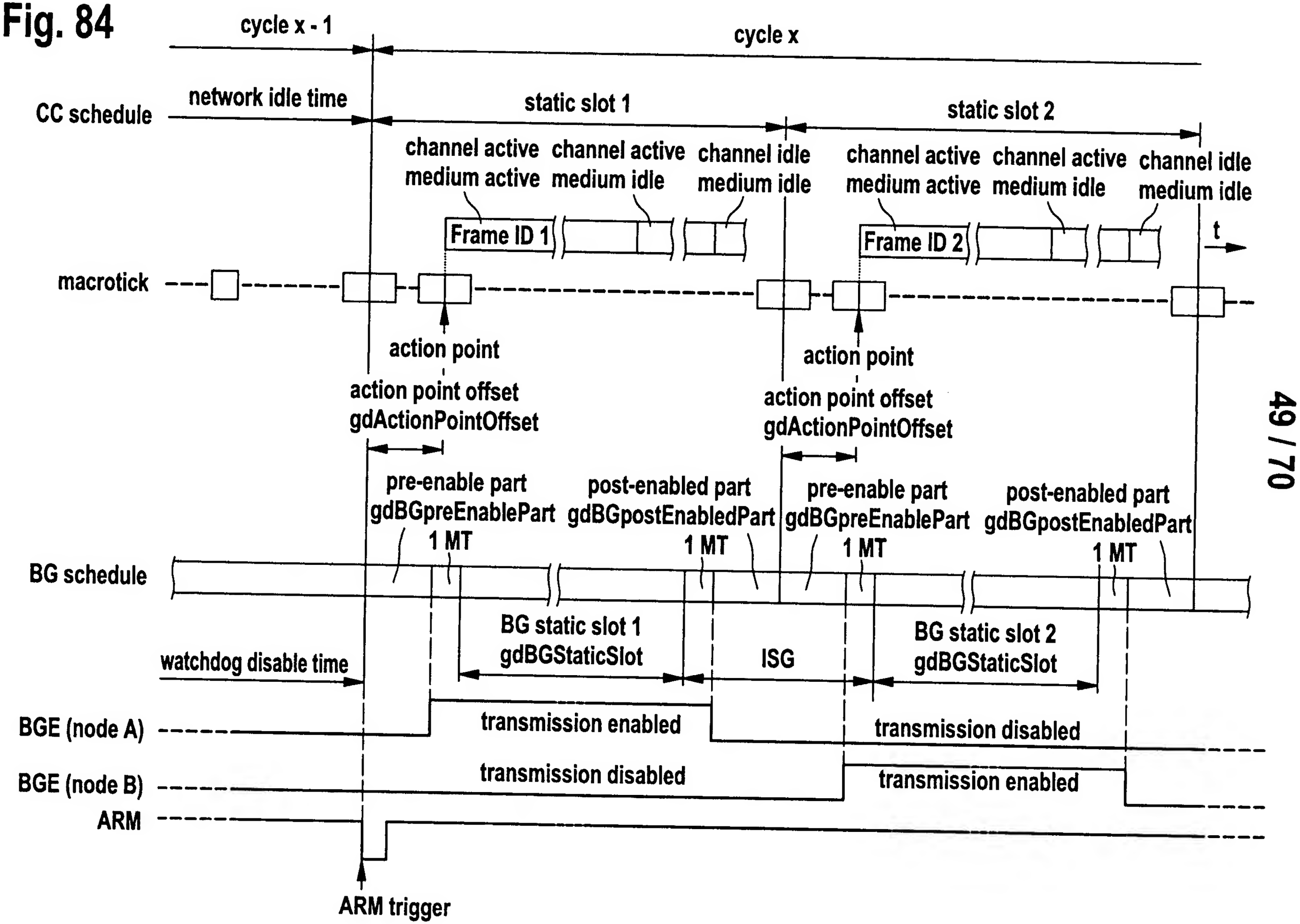

time base
derived
trigger
t
CC
communication
cycle x - 1
static segment
dynamic segment
network idle time
communication cycle x
symbol window
communication
cycle x + 1
BG
BG static segment
BG dynamic segment
BG symbol window
BG watchdog
disable time

Fig. 83

Fig. 84


cycle x - 1
cycle x
CC schedule
network idle time
static slot 1
static slot 2
channel active
medium active
channel active
medium idle
channel idle
medium idle
channel active
medium active
channel active
medium idle
channel idle
medium idle
Frame ID 1
Frame ID 2
t
macrotick
action point
action point offset
gdActionPointOffset
action point
action point offset
gdActionPointOffset
pre-enable part
gdBGpreEnablePart
1 MT
post-enabled part
gdBGpostEnabledPart
1 MT
pre-enable part
gdBGpreEnablePart
1 MT
post-enabled part
gdBGpostEnabledPart
1 MT
BG schedule
BG static slot 1
gdBGStaticSlot
ISG
BG static slot 2
gdBGStaticSlot
watchdog disable time
BGE (node A)
transmission enabled
transmission disabled
BGE (node B)
transmission disabled
transmission enabled
ARM
ARM trigger

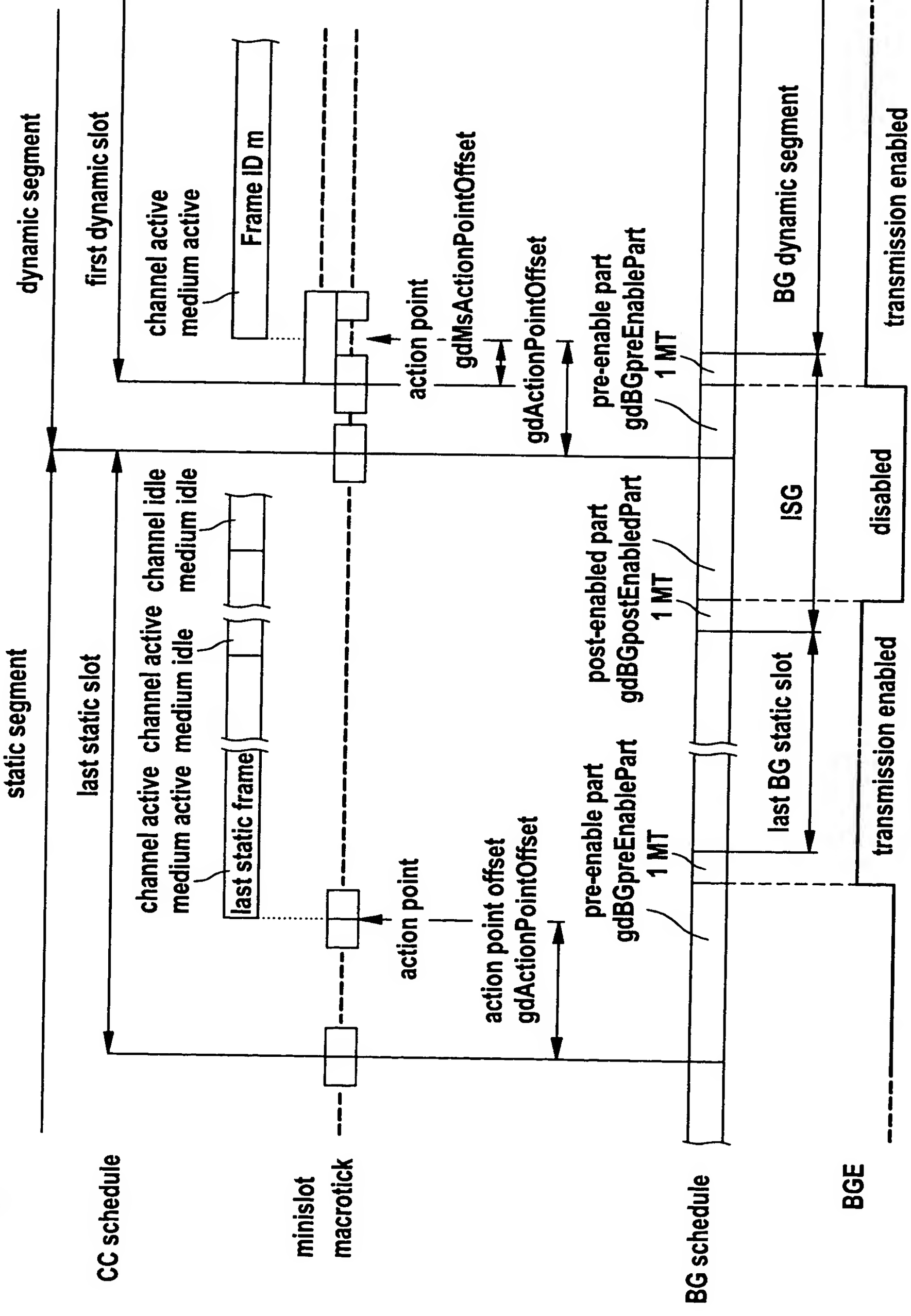

dynamic segment
first dynamic slot
channel active
medium active
Frame ID m
action point
gdMsActionPointOffset
gdActionPointOffset
pre-enable part
gdBGpreEnabledPart
1 MT
post-enabled part
gdBGpostEnabledPart
1 MT
BG dynamic segment
ISG
last BG static slot
transmission enabled
disabled
static segment
last static slot
channel active channel idle
medium active medium idle
last static frame
channel active channel idle
medium active medium idle
last static frame
action point
action point offset
gdActionPointOffset
pre-enable part
gdBGpreEnabledPart
1 MT
CC schedule
minislot
macrotick
BG schedule
BGE

Fig. 85

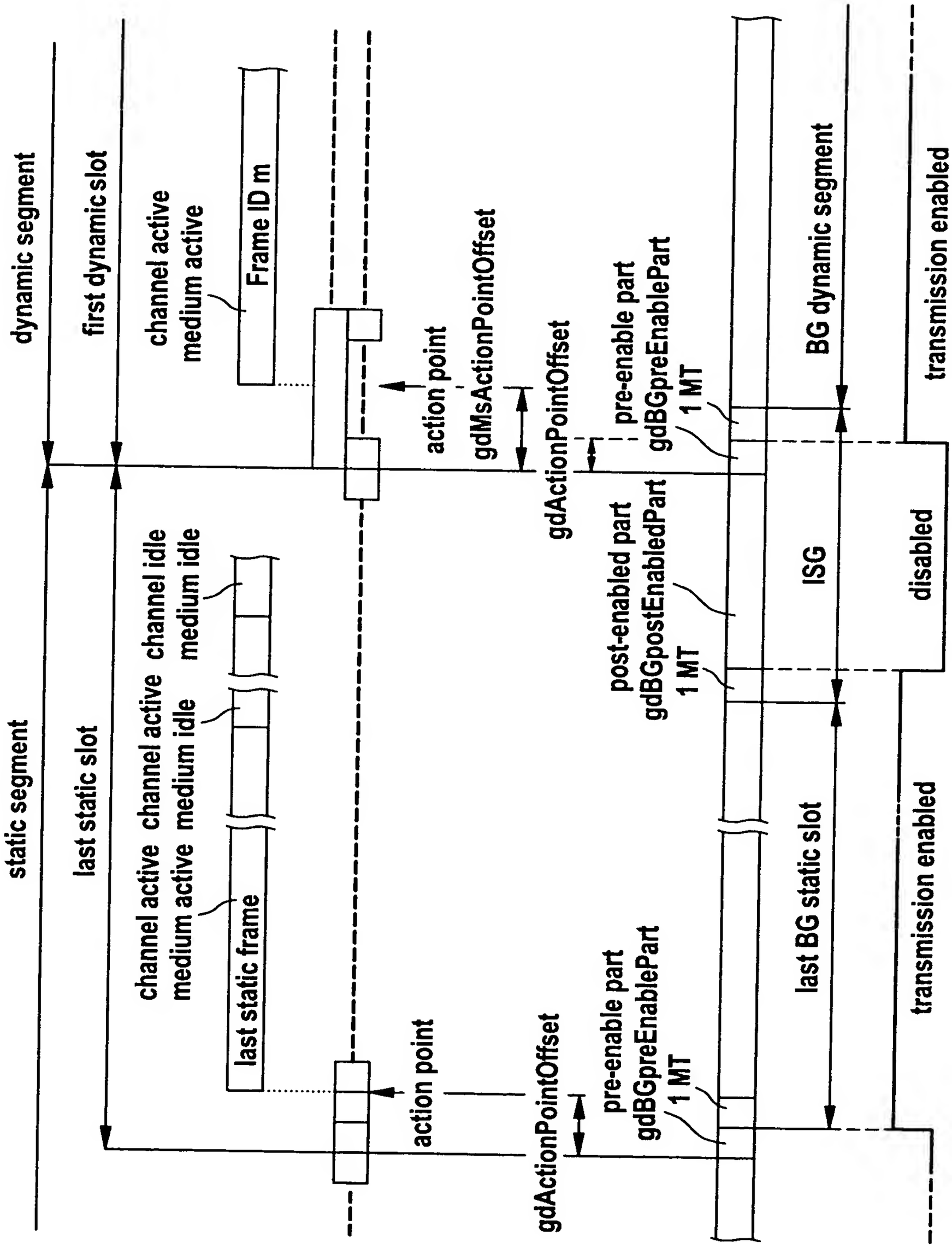

dynamic segment
first dynamic slot
channel active
medium active
Frame ID m
action point
gdMsActionPointOffset
gdActionPointOffset
pre-enable part
gdBGpreEnablePart
1 MT
BG dynamic segment
transmission enabled
post-enabled part
gdBGpostEnabledPart
1 MT
ISG
disabled
static segment
last static slot
channel idle
medium idle
channel active
medium idle
channel idle
medium active
last static frame
action point
gdActionPointOffset
pre-enable part
gdBGpreEnablePart
1 MT
last BG static slot
transmission enabled
CC schedule
minislot
macrotick
BG schedule
BGE

Fig. 86

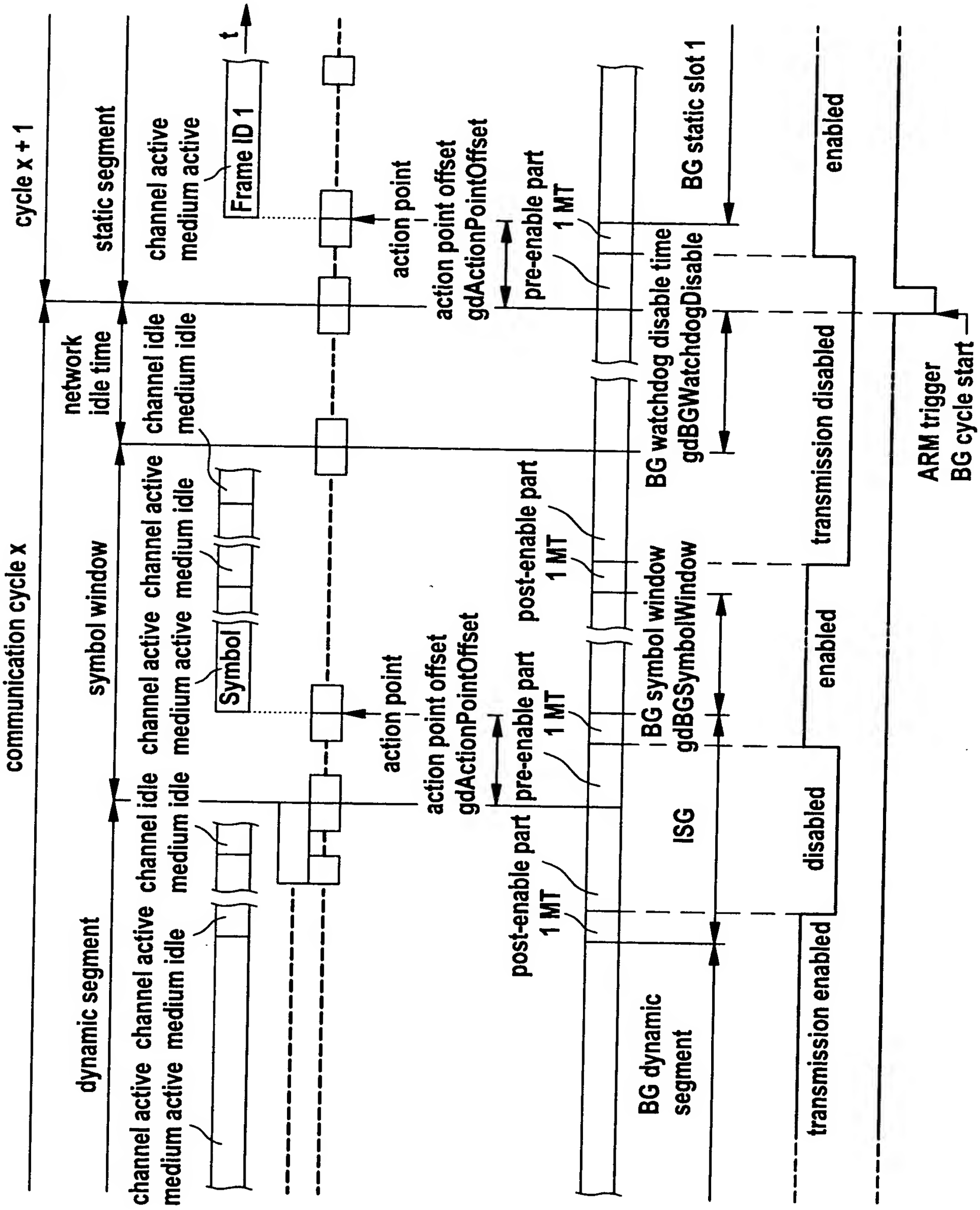

communication cycle x
cycle x + 1
symbol window
network idle time
static segment
dynamic segment
channel active
medium active
channel idle
medium idle
Frame ID 1
Symbol
t
action point
action point offset
gdActionPointOffset
pre-enable part
post-enable part
1 MT
BG symbol window
ISG
gdBGSymbolWindow
BG static slot 1
BG watchdog disable time
gdBGWatchdogDisable
BG watchdog
enabled
disabled
transmission disabled
transmission enabled
ARM trigger
BG cycle start
CC schedule
minislot
macrotick
BG schedule
BG dynamic segment
BGE

Fig. 87

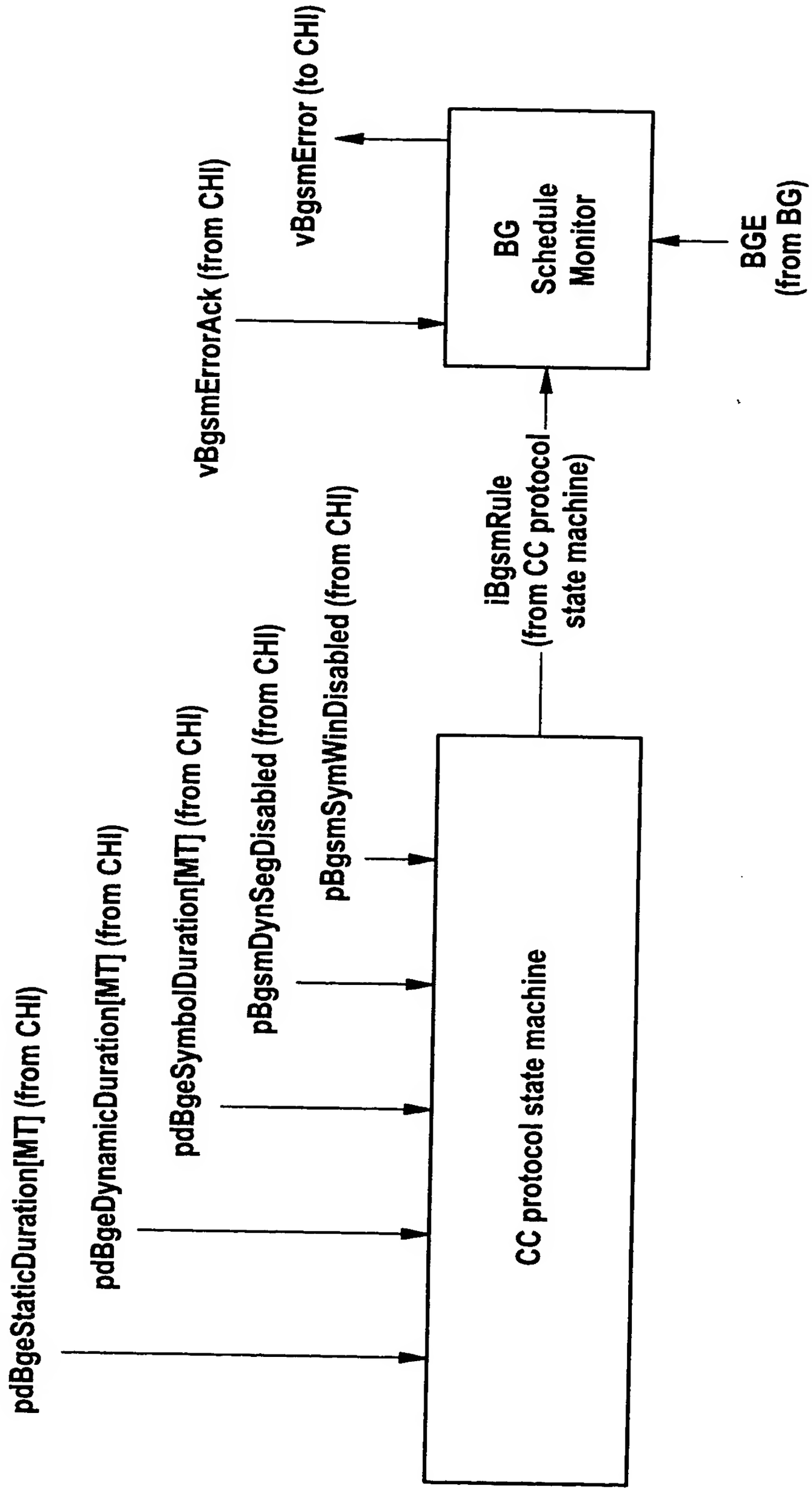
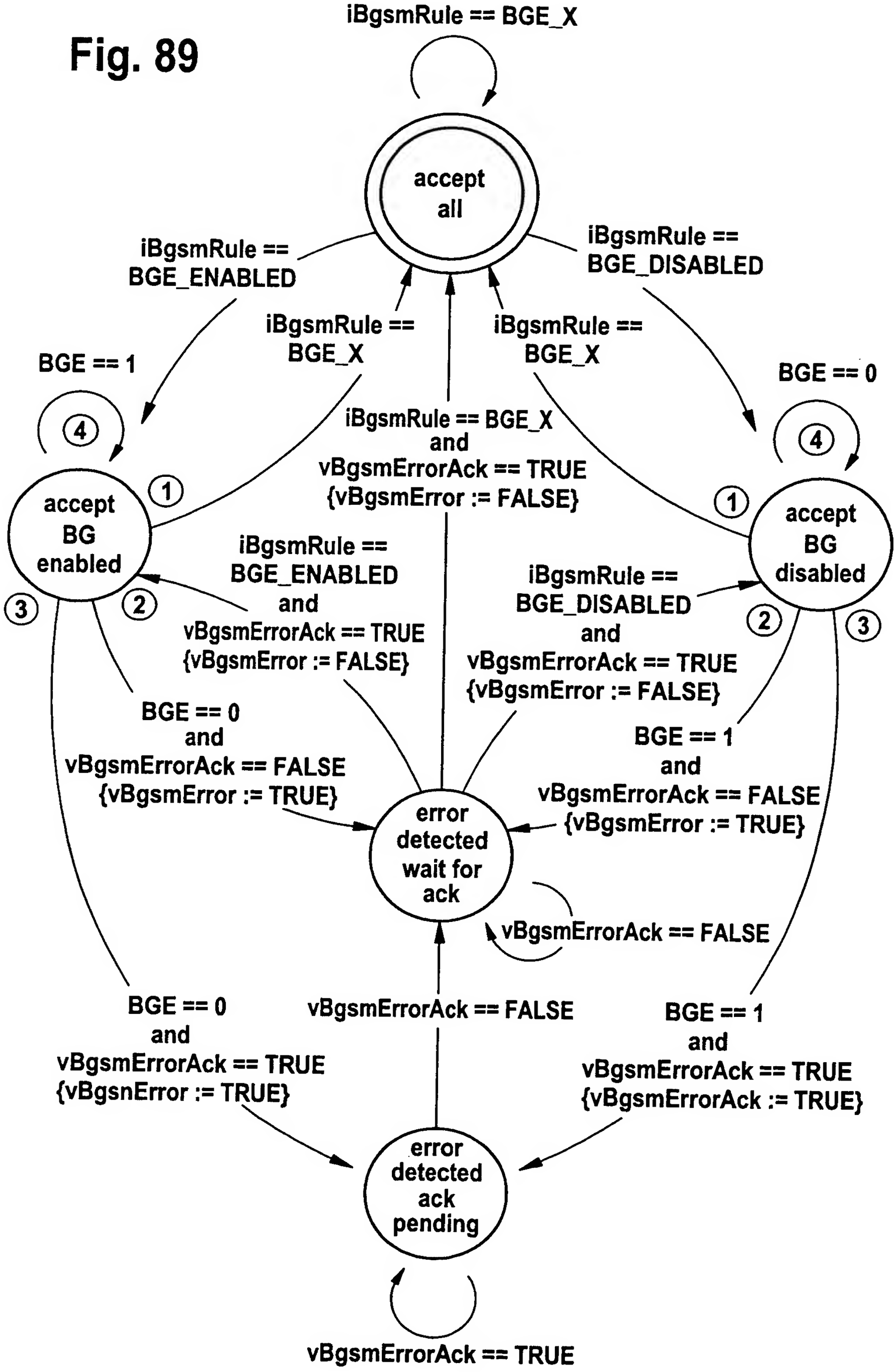

vBgsmError (to CHI)
BG
Schedule
Monitor
BGE
(from BG)
vBgsmErrorAck (from CHI)
iBgsmRule
(from CC protocol
state machine)
pBgsmSymWinDisabled (from CHI)
pBgsmDynSegDisabled (from CHI)
pdBgeSymbolDuration[MT] (from CHI)
pdBgeDynamicDuration[MT] (from CHI)
pdBgeStaticDuration[MT] (from CHI)
CC protocol state machine

Fig. 88

Fig. 89


iBgsmRule == BGE_X
accept all
iBgsmRule == BGE_ENABLED
iBgsmRule == BGE_X
iBgsmRule == BGE_X
iBgsmRule == BGE_DISABLED
BGE == 1
BGE == 0
4
4
iBgsmRule == BGE_X and vBgsmErrorAck == TRUE {vBgsmError := FALSE}
accept BG enabled
accept BG disabled
1
1
iBgsmRule == BGE_ENABLED and vBgsmErrorAck == TRUE {vBgsmError := FALSE}
iBgsmRule == BGE_DISABLED and vBgsmErrorAck == TRUE {vBgsmError := FALSE}
3
2
2
3
BGE == 0 and vBgsmErrorAck == FALSE {vBgsmError := TRUE}
BGE == 1 and vBgsmErrorAck == FALSE {vBgsmError := TRUE}
error detected wait for ack
vBgsmErrorAck == FALSE
BGE == 0 and vBgsmErrorAck == TRUE {vBgsnError := TRUE}
vBgsmErrorAck == FALSE
BGE == 1 and vBgsmErrorAck == TRUE {vBgsmErrorAck := TRUE}
error detected ack pending
vBgsmErrorAck == TRUE

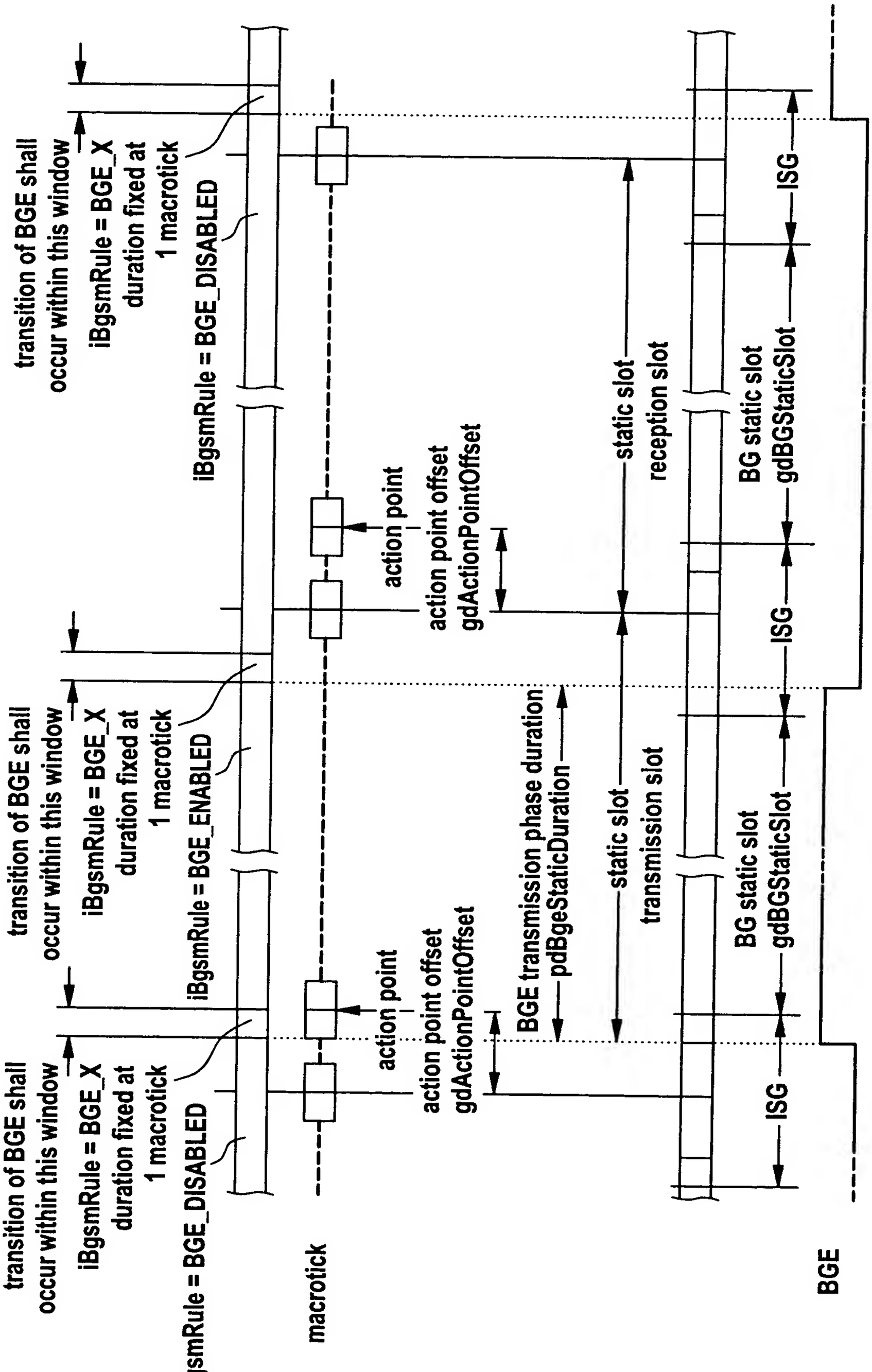

transition of BGE shall occur within this window
iBgsmRule = BGE_X duration fixed at 1 macrotick
iBgsmRule = BGE_DISABLED
iBgsmRule = BGE_ENABLED
macrotick
action point
action point offset
gdActionPointOffset
BGE transmission phase duration
pdBgeStaticDuration
static slot
transmission slot
reception slot
BG static slot
gdBGStaticSlot
ISG
BGE

Fig. 90

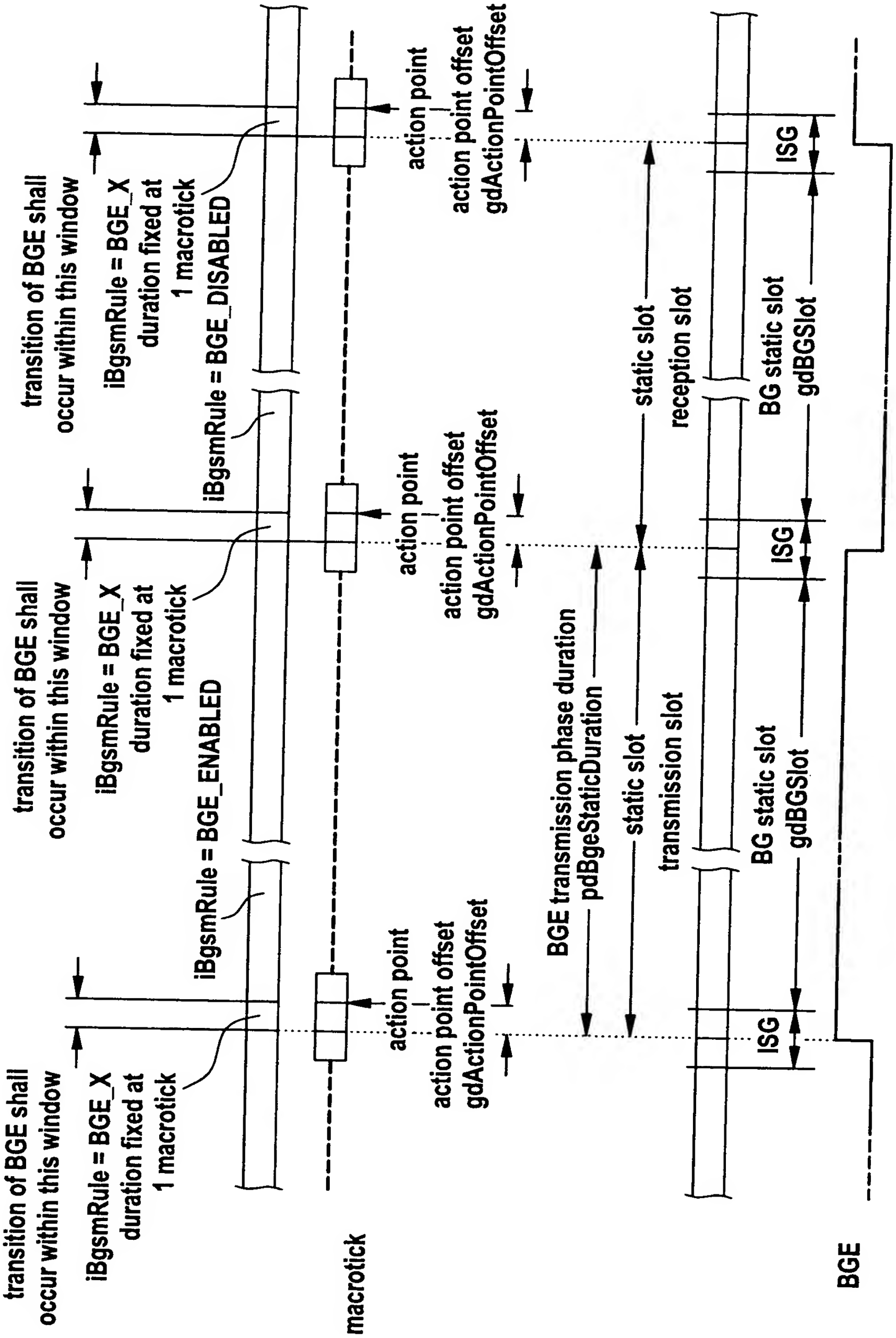

transition of BGE shall occur within this window
iBgsmRule = BGE_X duration fixed at 1 macrotick
iBgsmRule = BGE_DISABLED
iBgsmRule = BGE_ENABLED
action point
action point offset
gdActionPointOffset
BGE transmission phase duration
pdBgeStaticDuration
static slot
transmission slot
BG static slot
gdBGSlot
ISG
reception slot
macrotick
BGE

Fig. 91

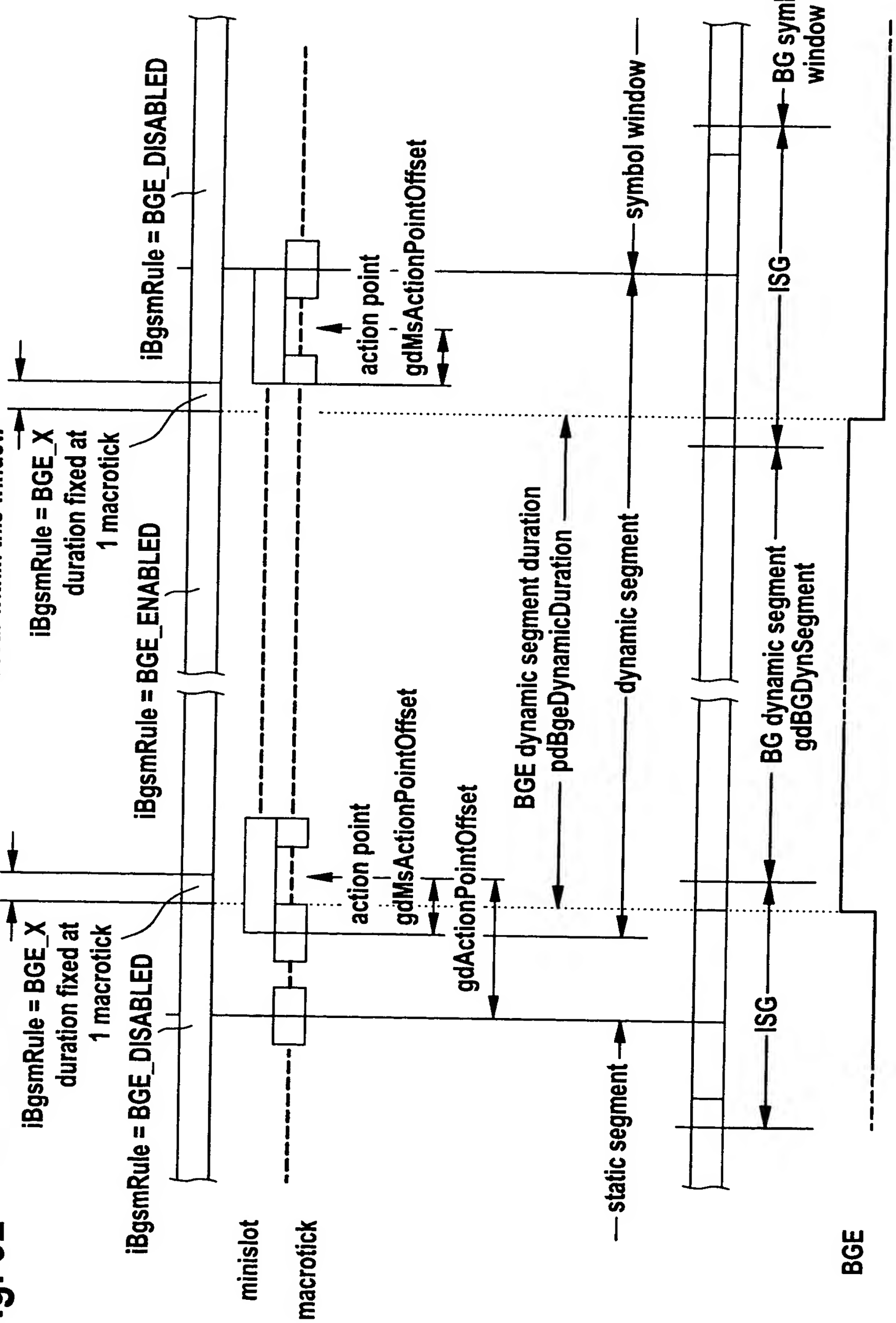
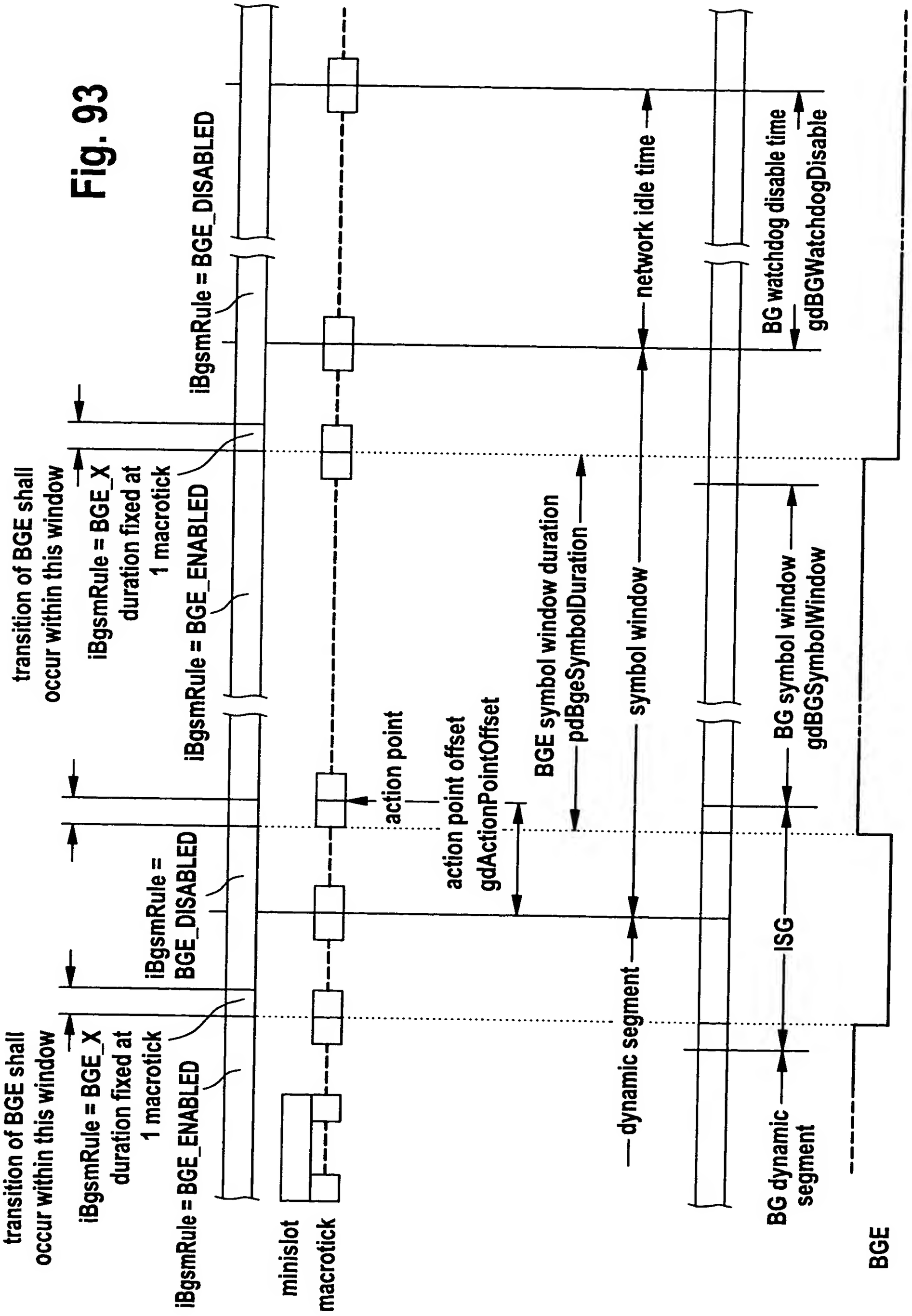

iBgsmRule = BGE_DISABLED
transition of BGE shall occur within this window
iBgsmRule = BGE_X duration fixed at 1 macrotick
iBgsmRule = BGE_ENABLED
action point
gdMsActionPointOffset
symbol window
BG symbol window
ISG
BGE dynamic segment duration
pdBgeDynamicDuration
dynamic segment
BG dynamic segment
gdBGDynSegment
gdMsActionPointOffset
gdActionPointOffset
static segment
minislot
macrotick
BGE

Fig. 92

Fig. 93



06071457EP

Fig. 94

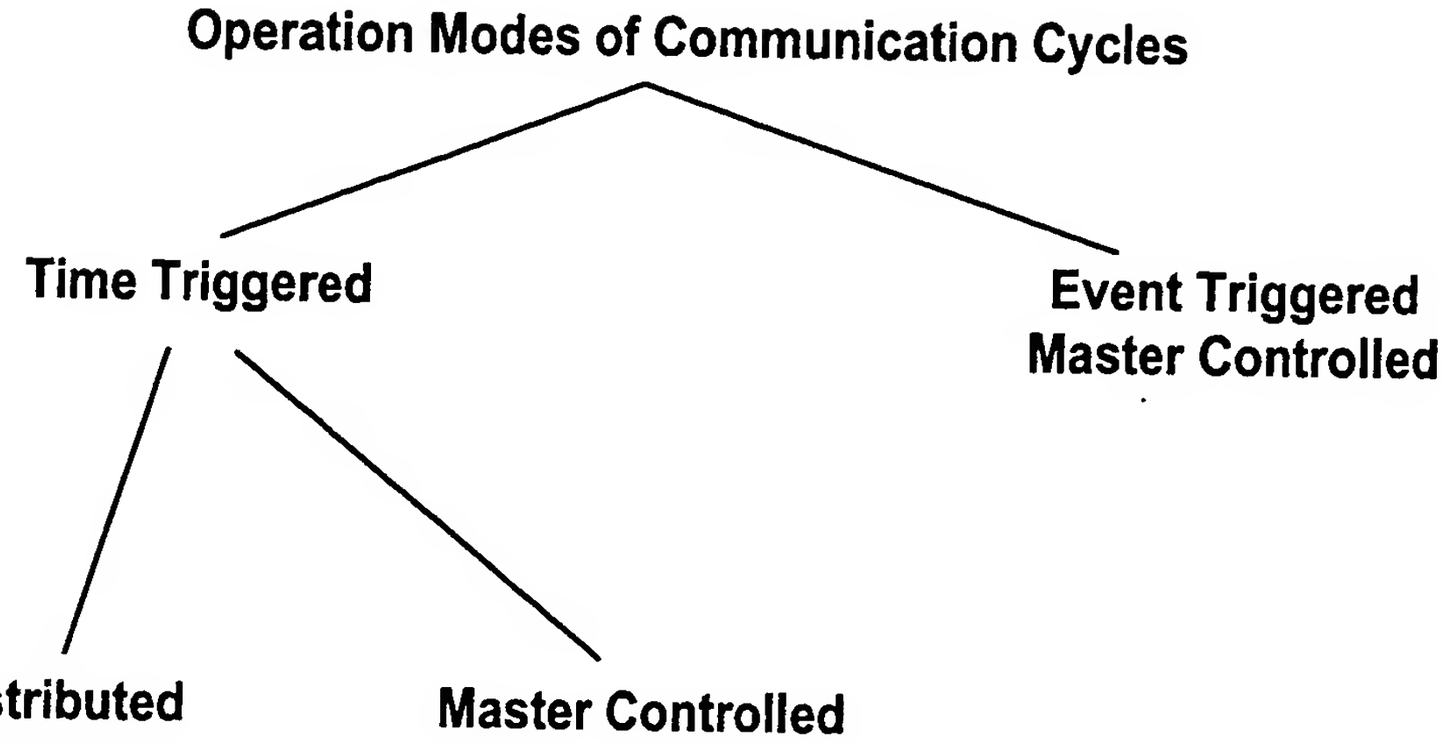
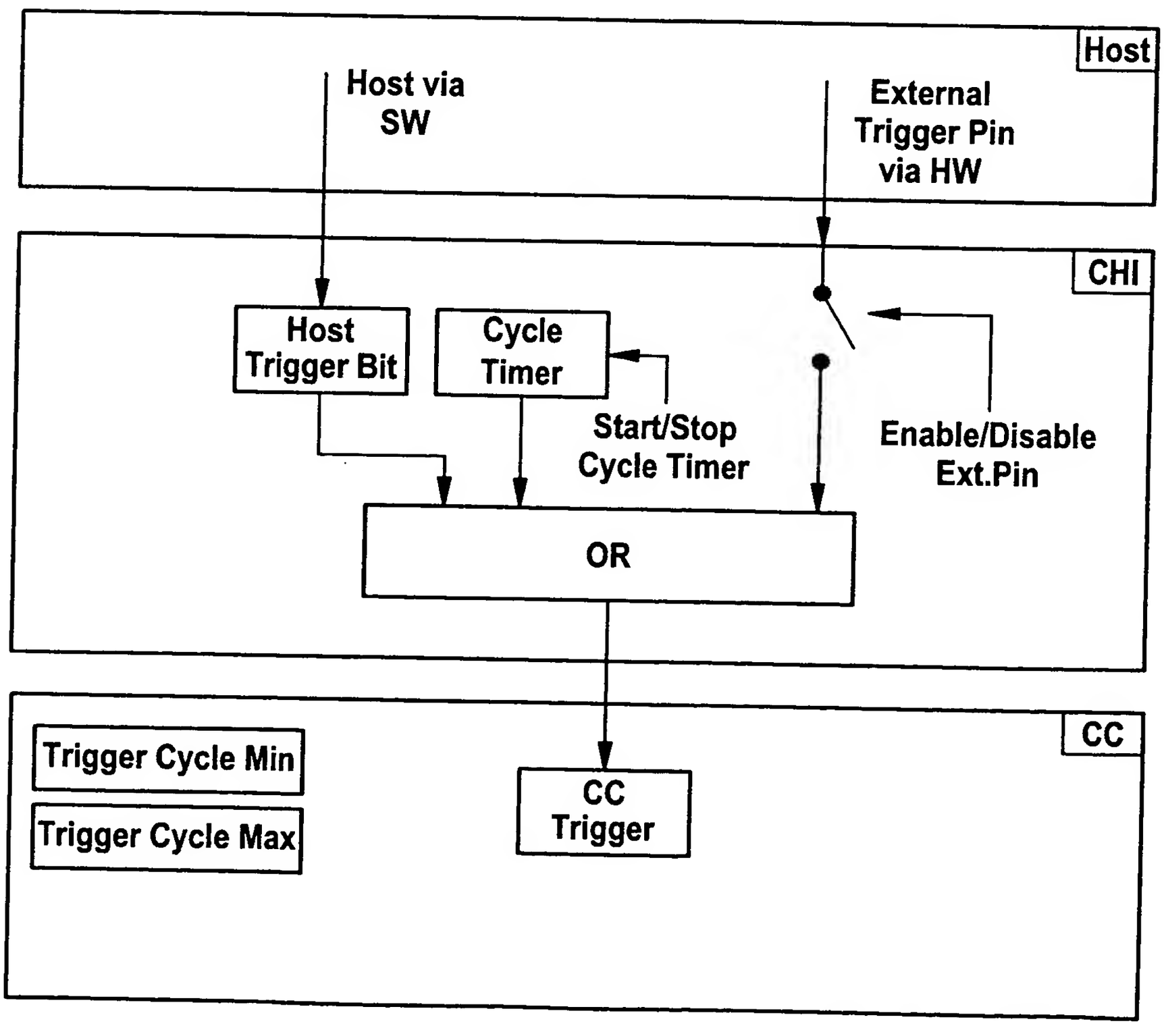

Operation Modes of Communication Cycles
Time Triggered
Event Triggered
Master Controlled
Distributed
Master Controlled

Fig. 95


Host
Host via SW
External Trigger Pin via HW
CHI
Host Trigger Bit
Cycle Timer
Start/Stop Cycle Timer
Enable/Disable Ext.Pin
OR
CC
Trigger Cycle Min
Trigger Cycle Max
CC Trigger

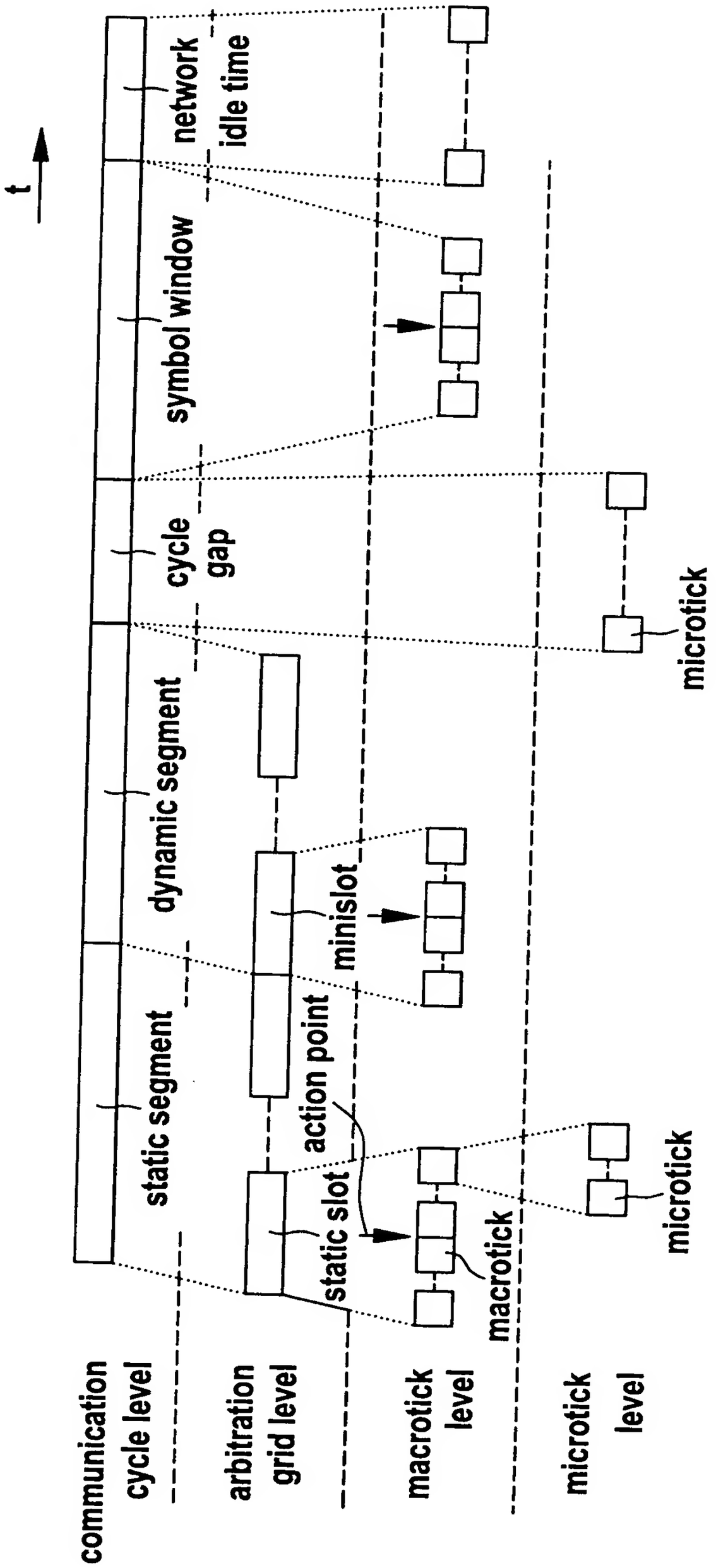

t
network idle time
symbol window
cycle gap
dynamic segment
minislot
action point
static segment
static slot
macrotick
microtick
communication cycle level
arbitration grid level
macrotick level
microtick level

Fig. 96

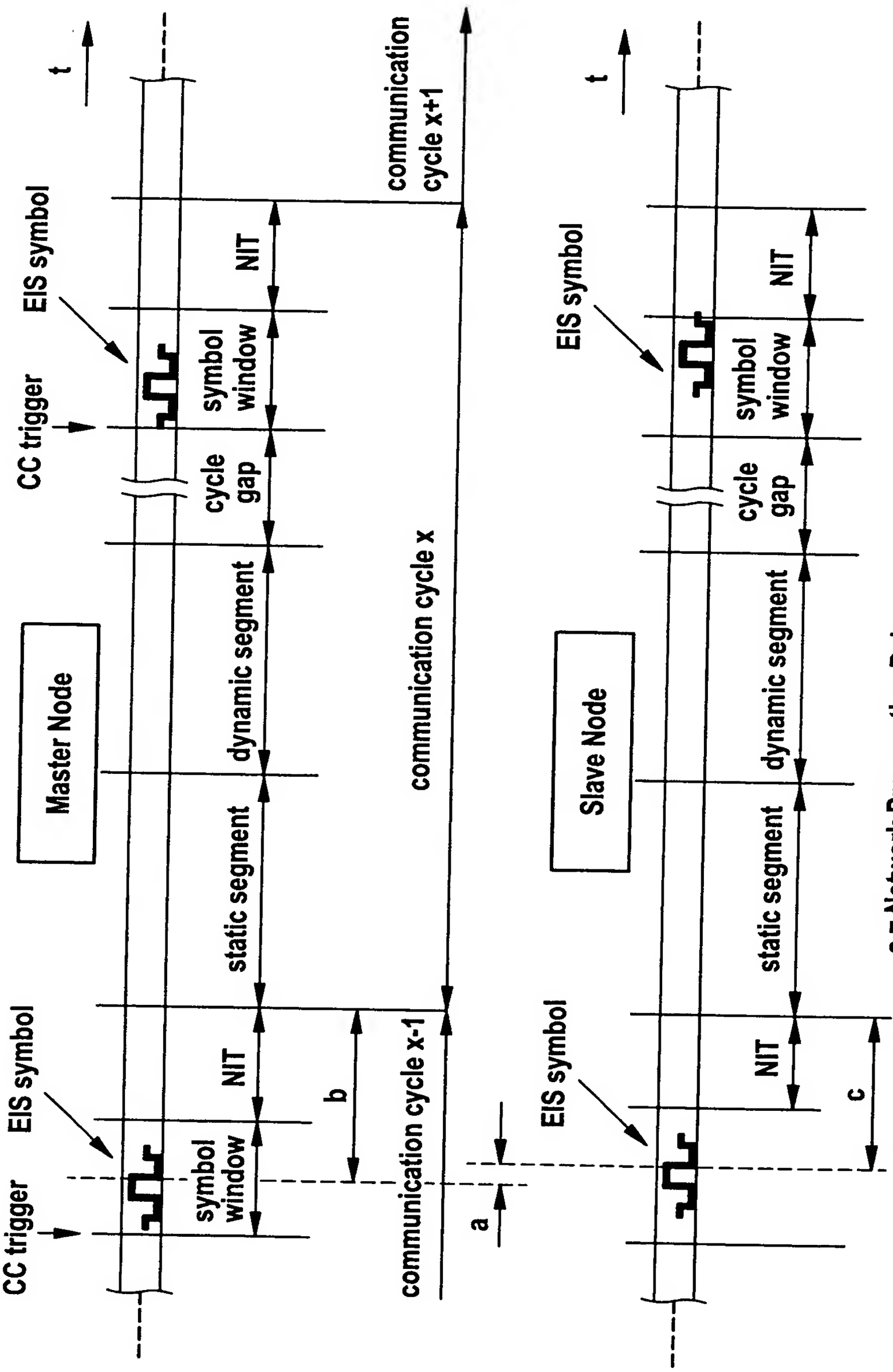

t
EIS symbol
NIT
symbol window
communication cycle x+1
CC trigger
cycle gap
Master Node
dynamic segment
static segment
communication cycle x
communication cycle x-1
Slave Node
a = Network Propagation Delay
b = pdMasterEIStoEndOfNit
c = pdSlaveEIStoEndOfNit

Fig. 97

Fig. 98

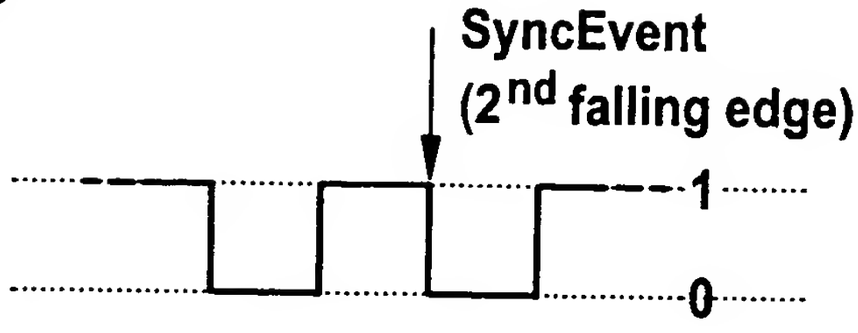

SyncEvent
(2nd falling edge)
1
0

Fig. 99

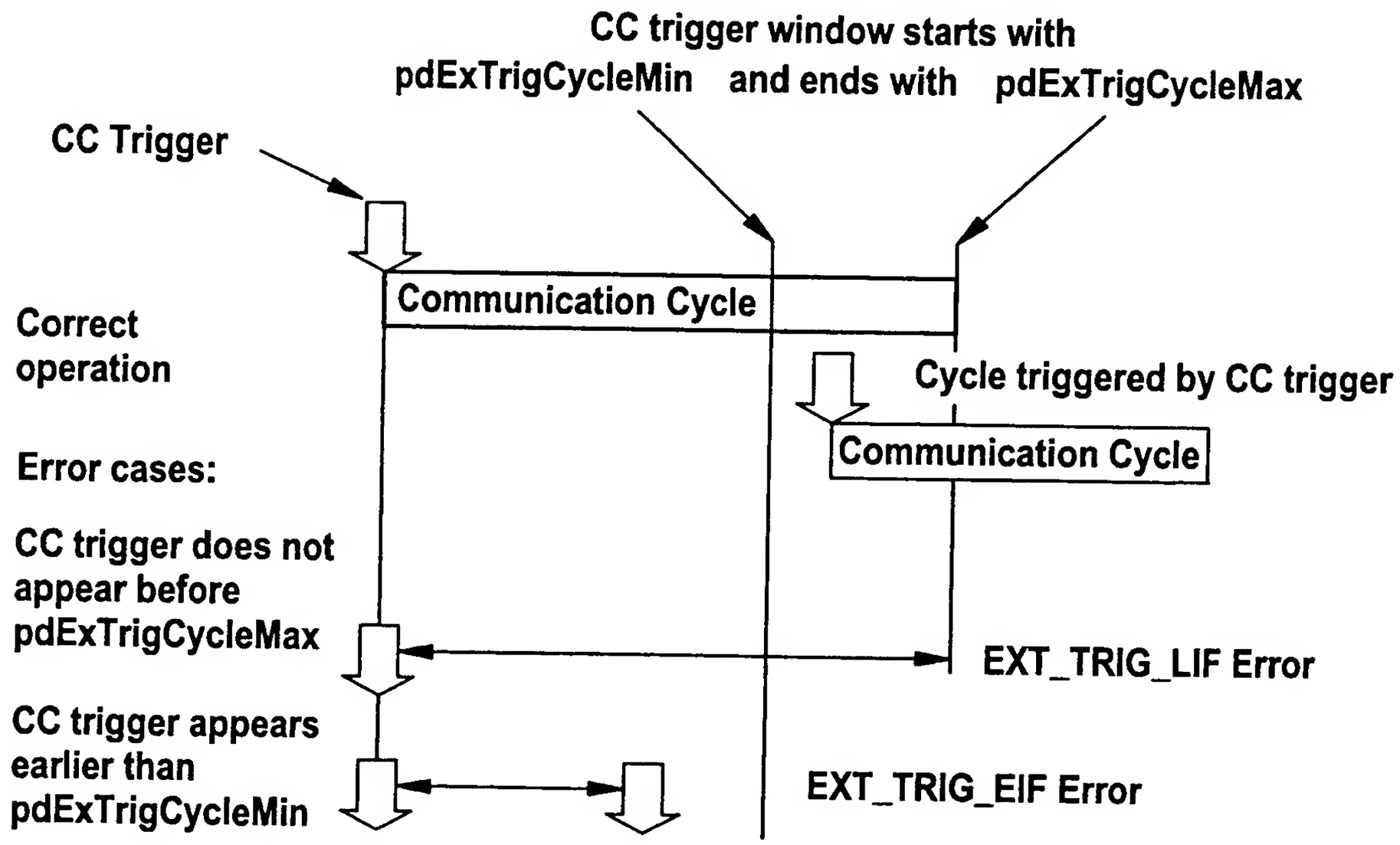

CC trigger window starts with
pdExTrigCycleMin and ends with pdExTrigCycleMax
CC Trigger
Communication Cycle
Correct operation
Cycle triggered by CC trigger
Communication Cycle
Error cases:
CC trigger does not appear before pdExTrigCycleMax
EXT_TRIG_LIF Error
CC trigger appears earlier than pdExTrigCycleMin
EXT_TRIG_EIF Error

Fig. 100

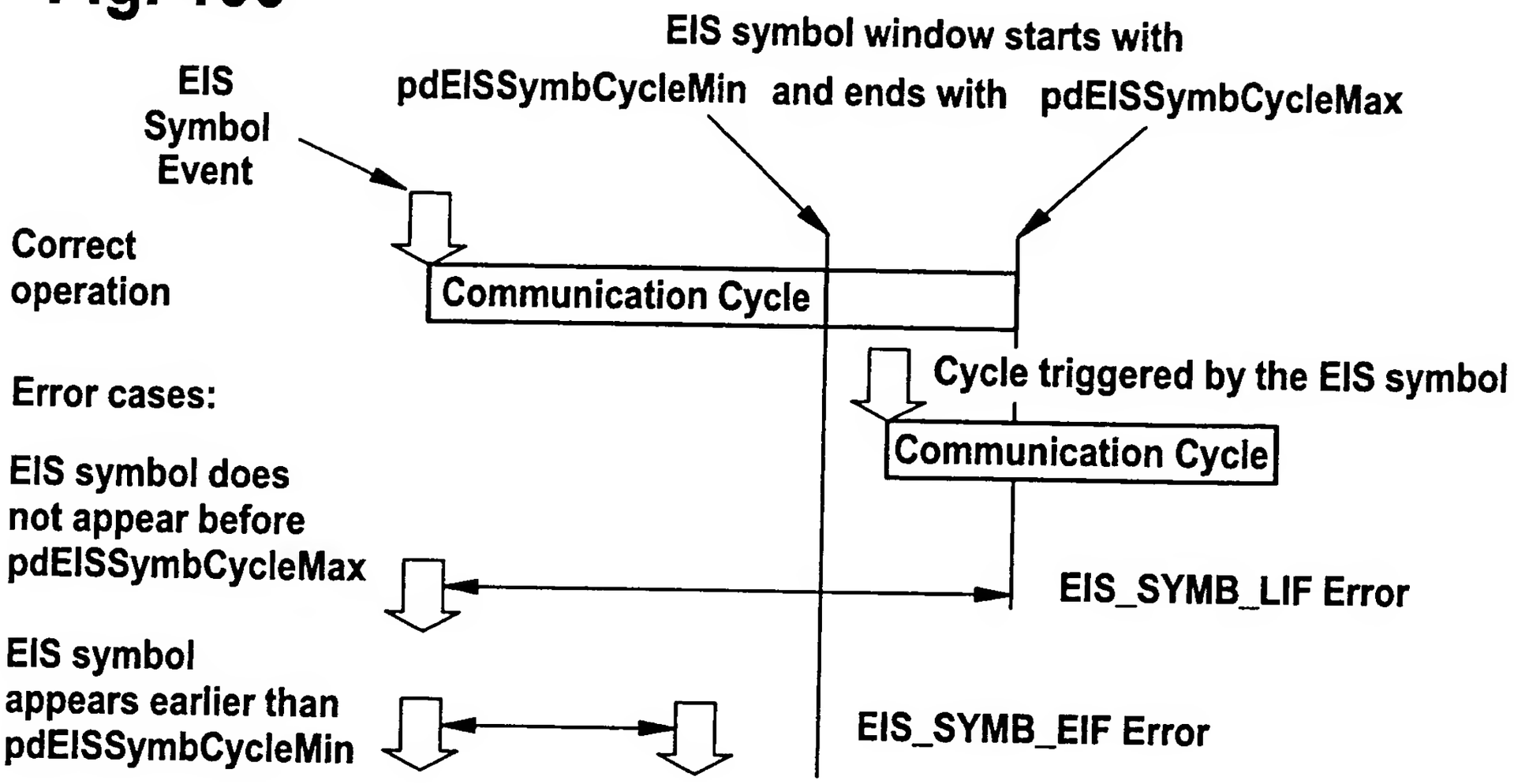


Fig. 101

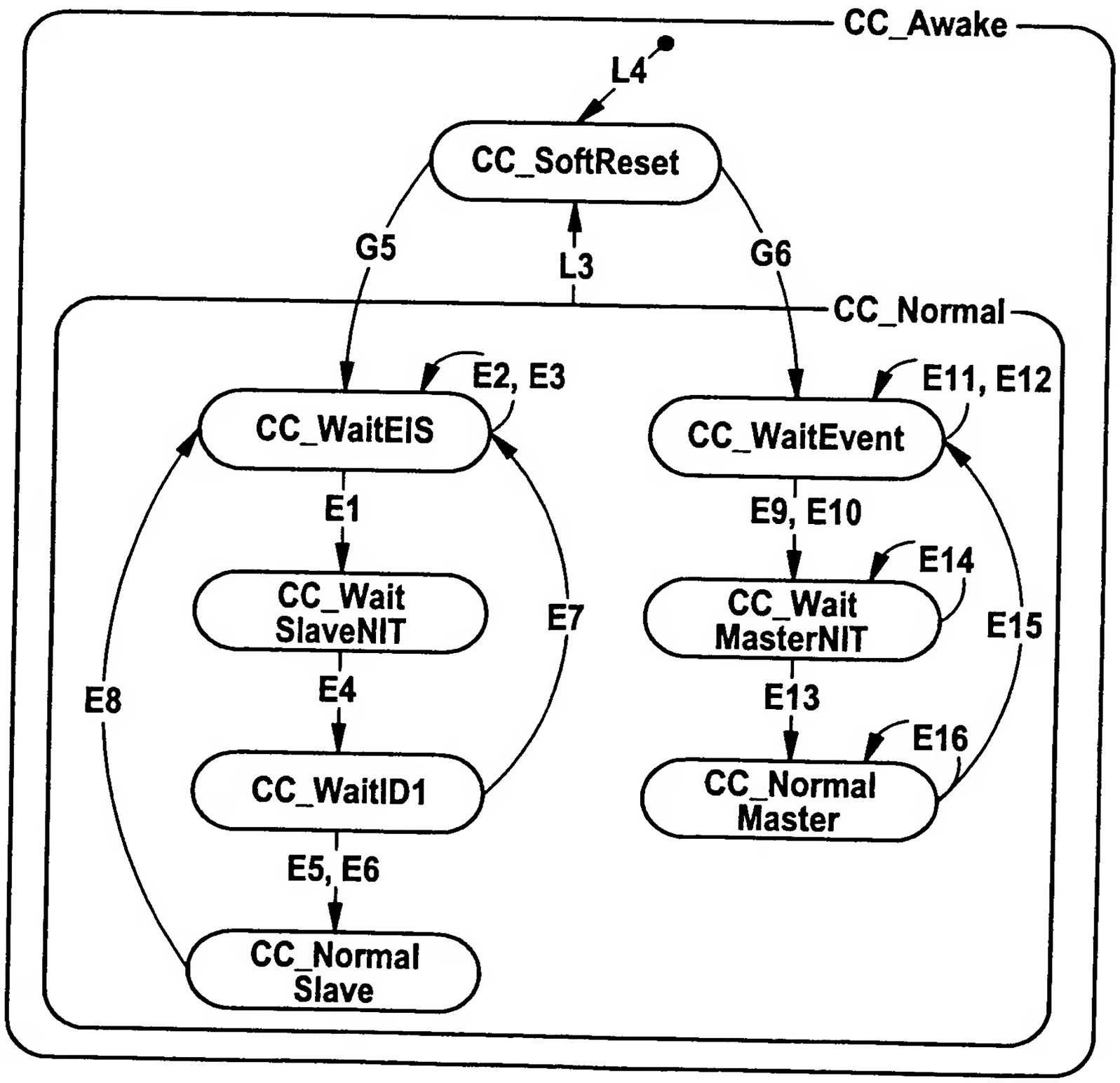


Fig. 102

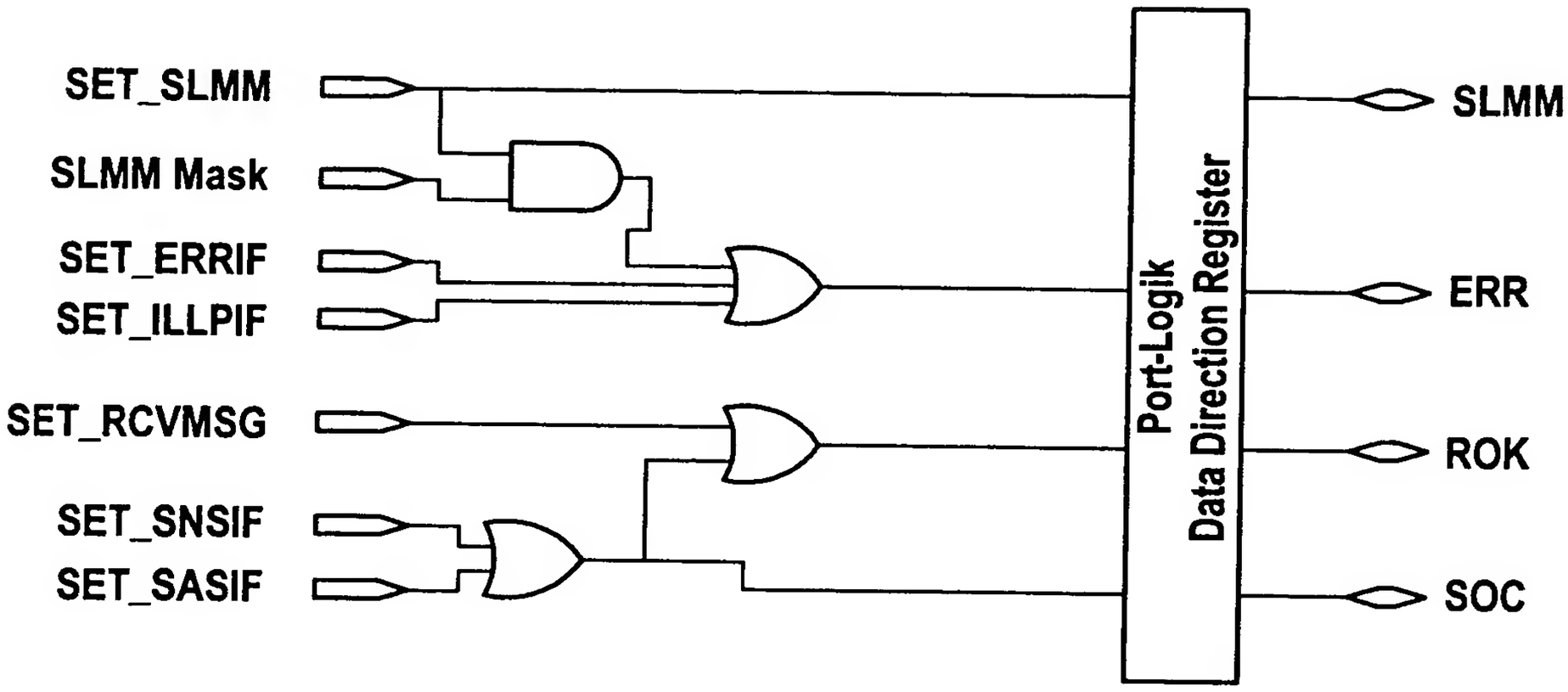


Fig. 103

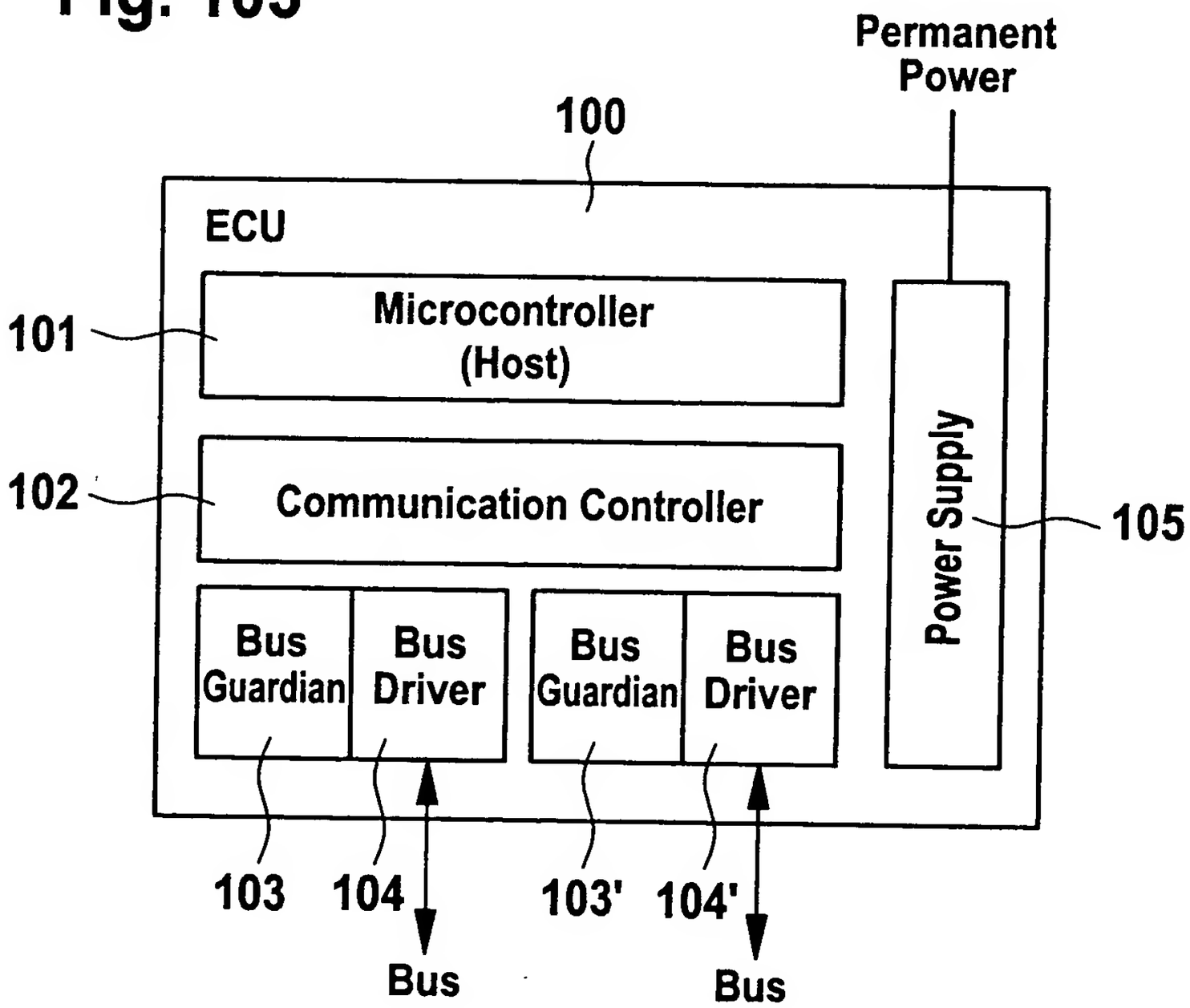


Fig. 104

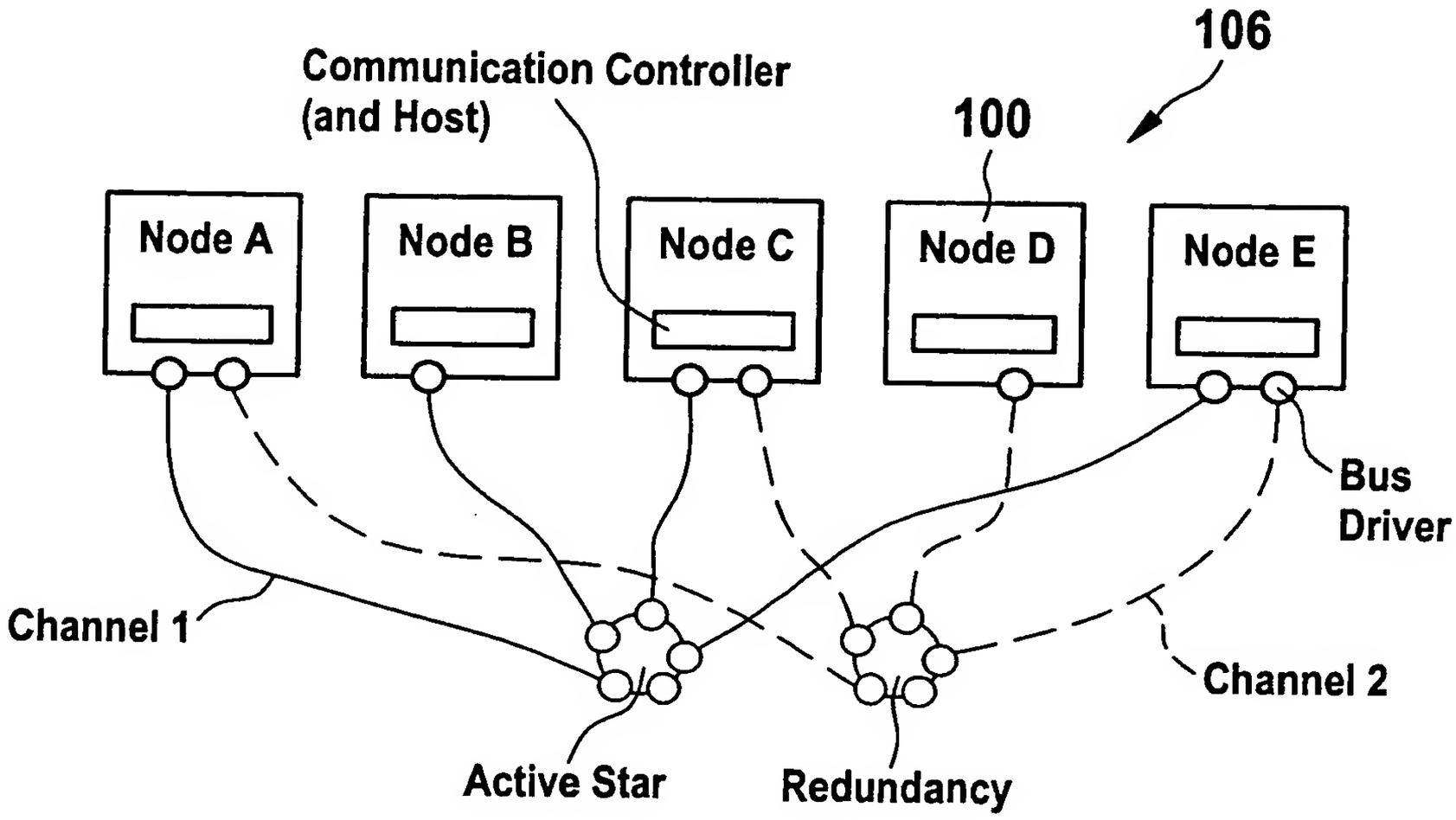


Fig. 105

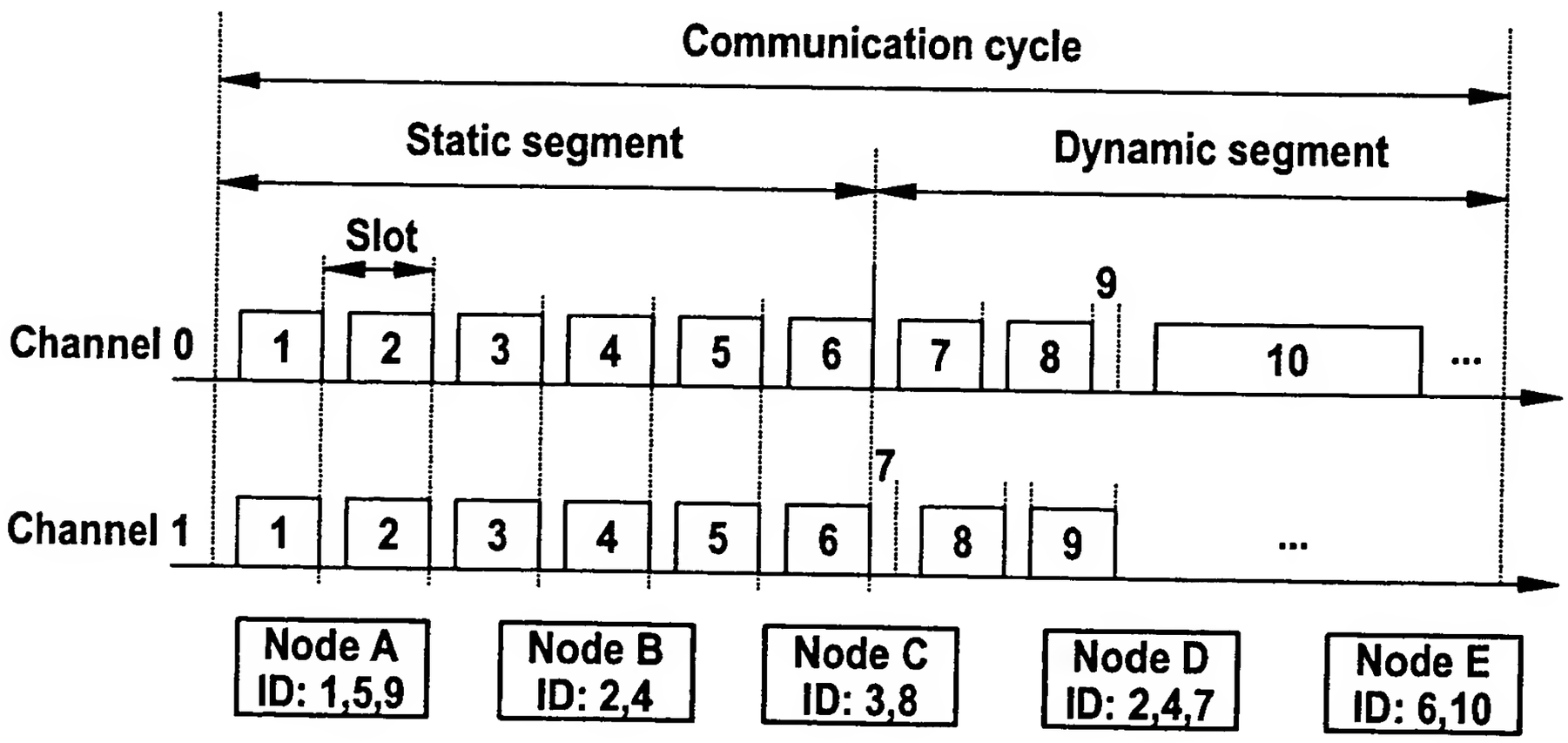


Fig. 106

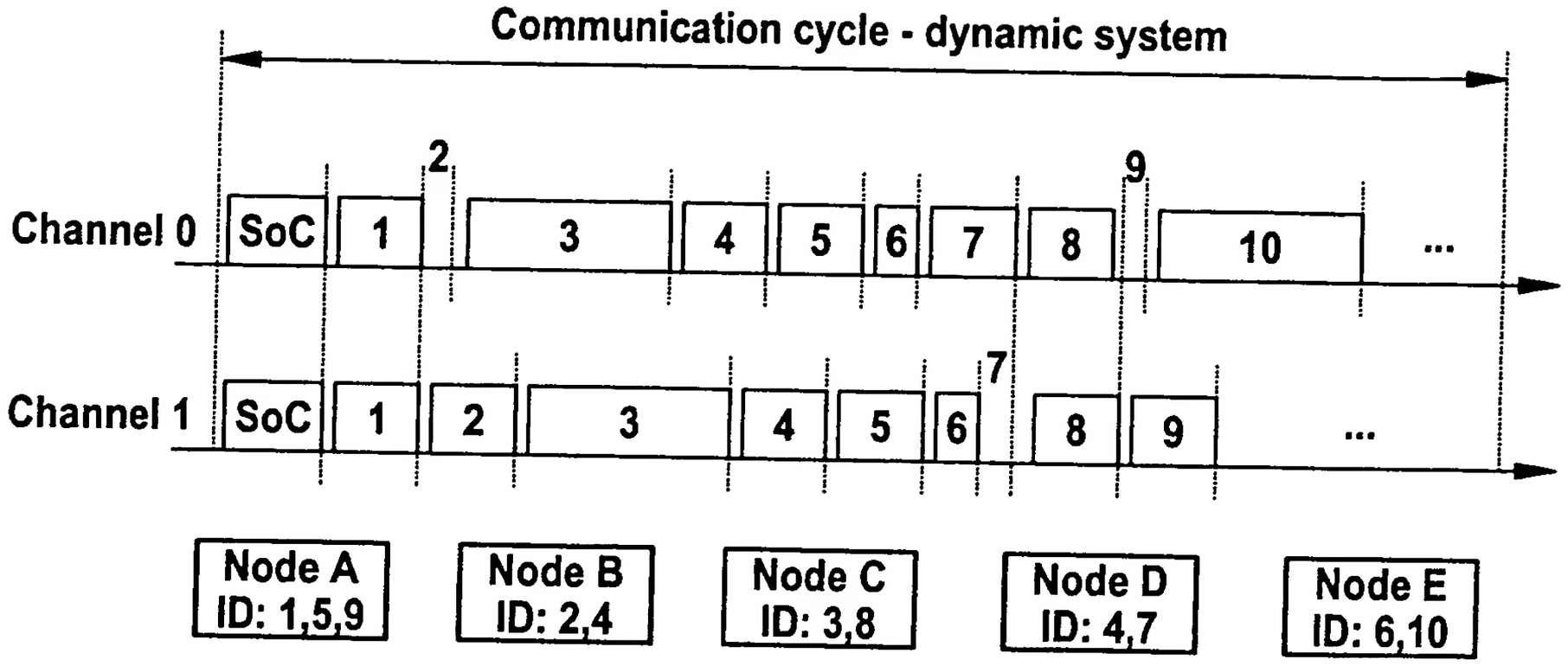


Fig. 107

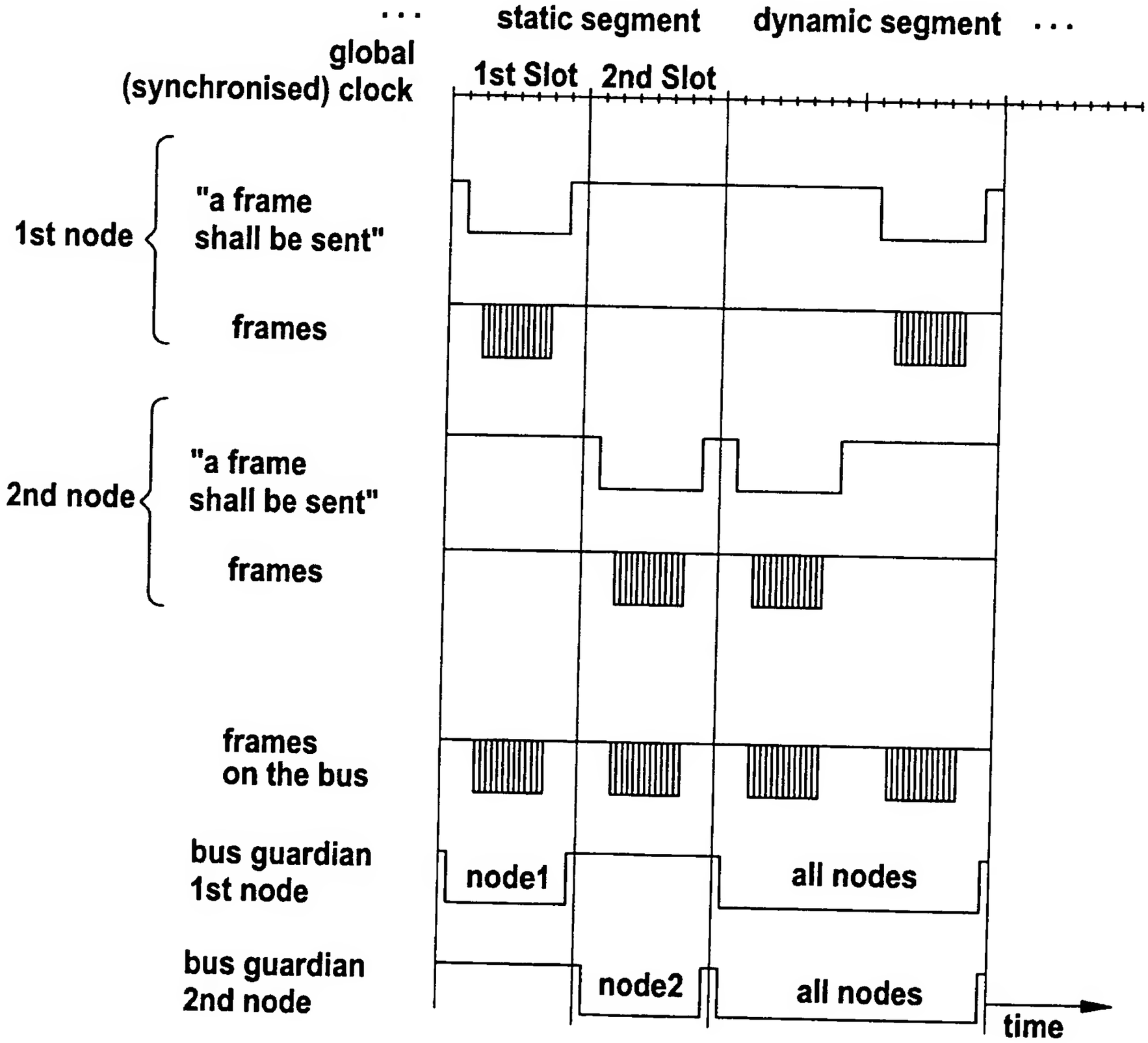


Fig. 108

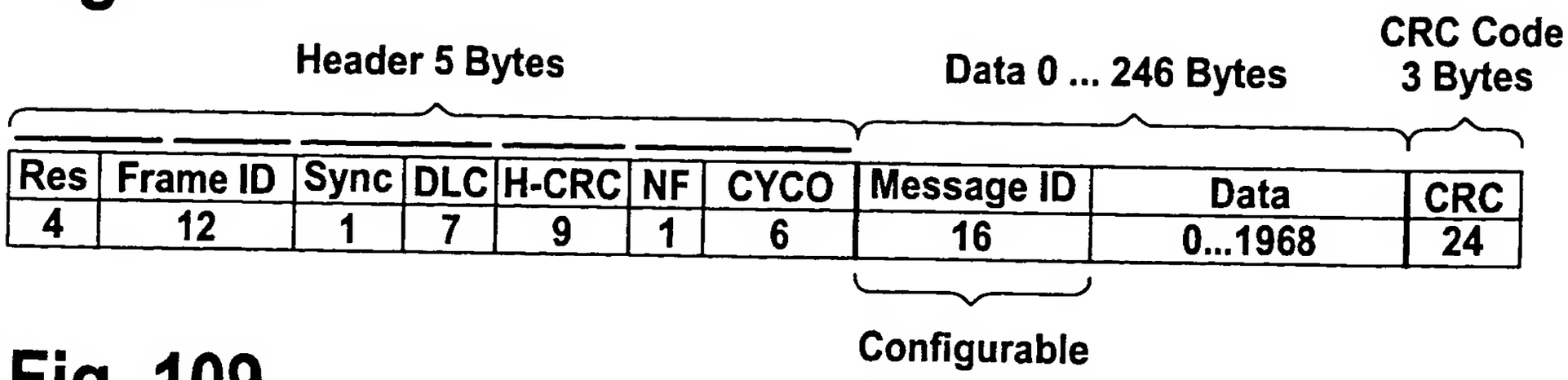


Fig. 109

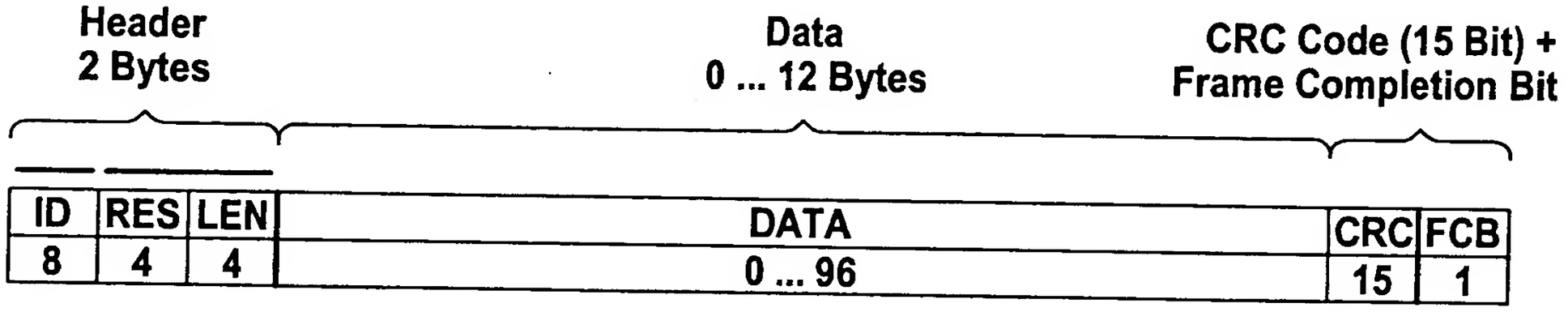


Fig. 110

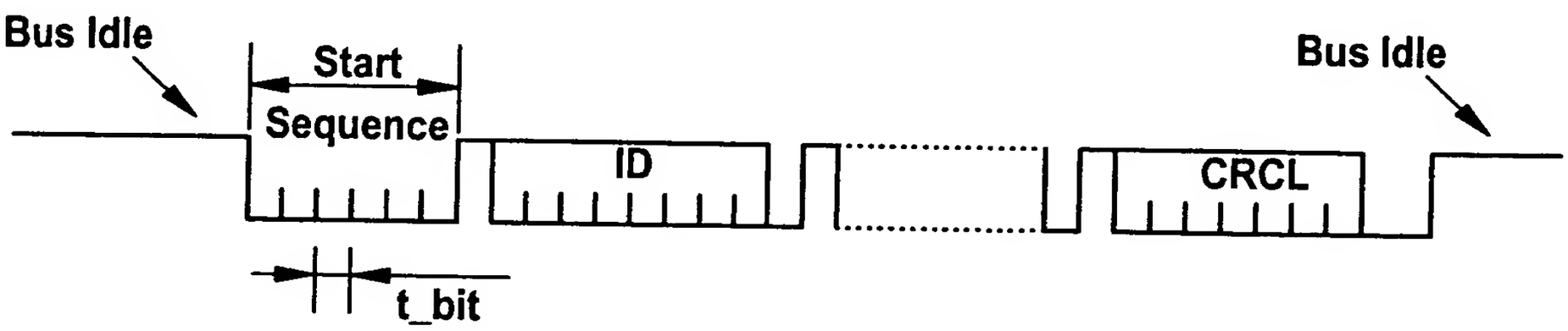


Fig. 111

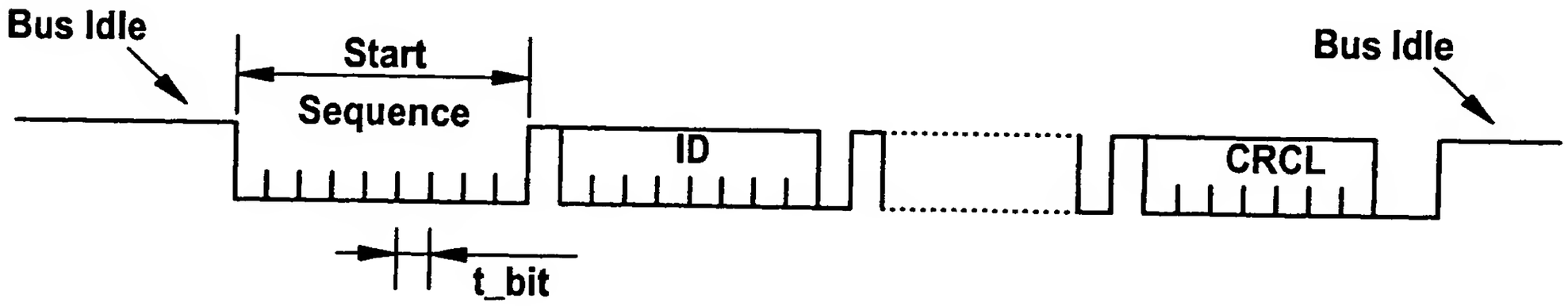


Fig. 112

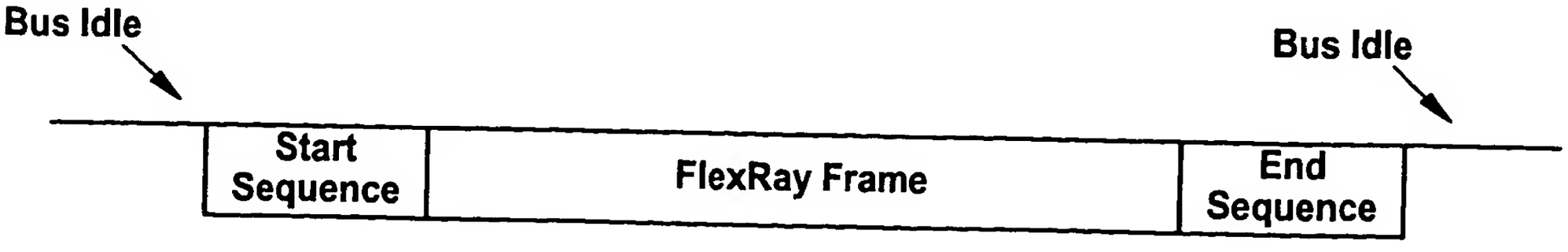


Fig. 113

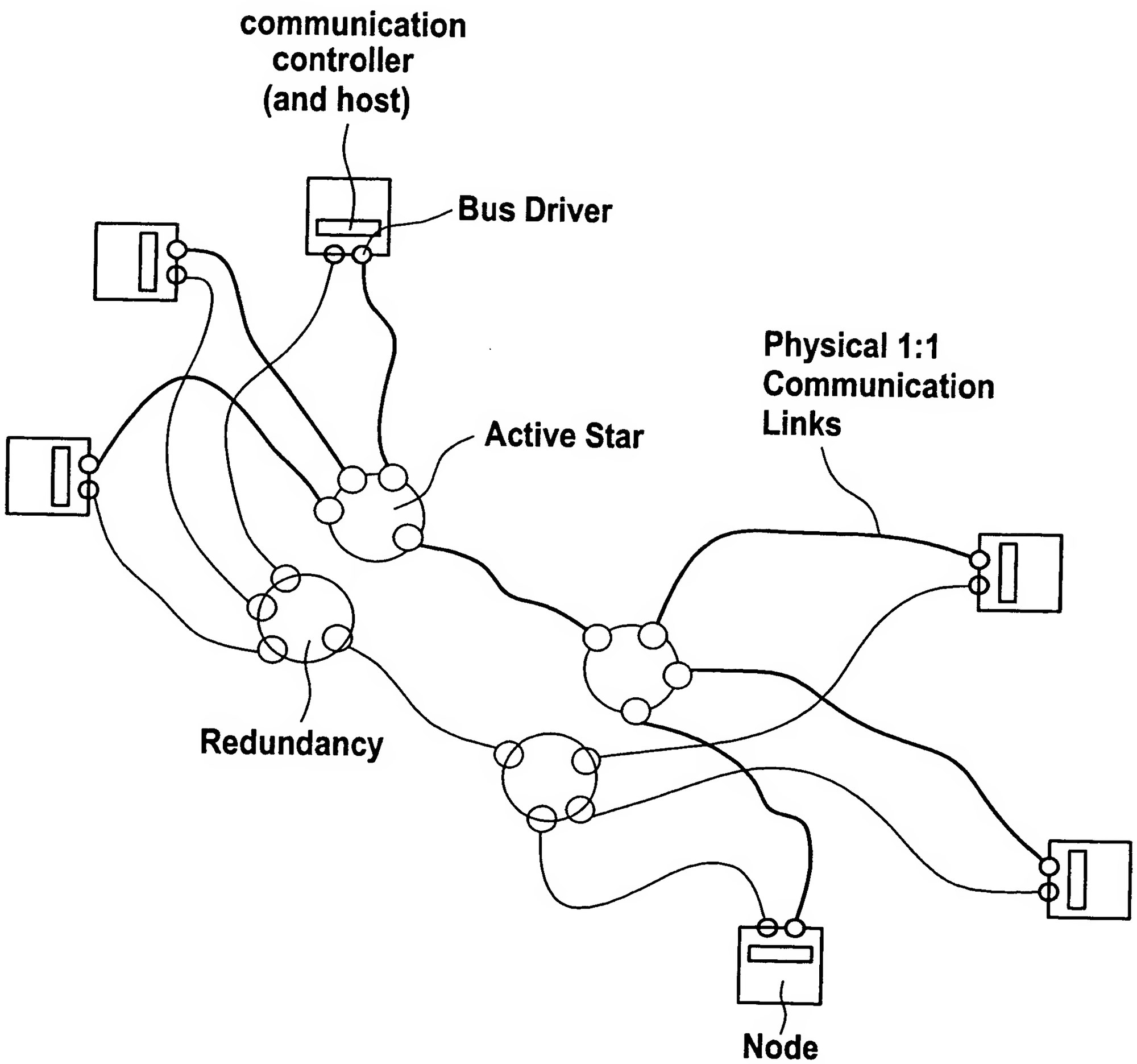

communication
controller
(and host)
Bus Driver
Active Star
Physical 1:1
Communication
Links
Redundancy
Node

Fig. 114

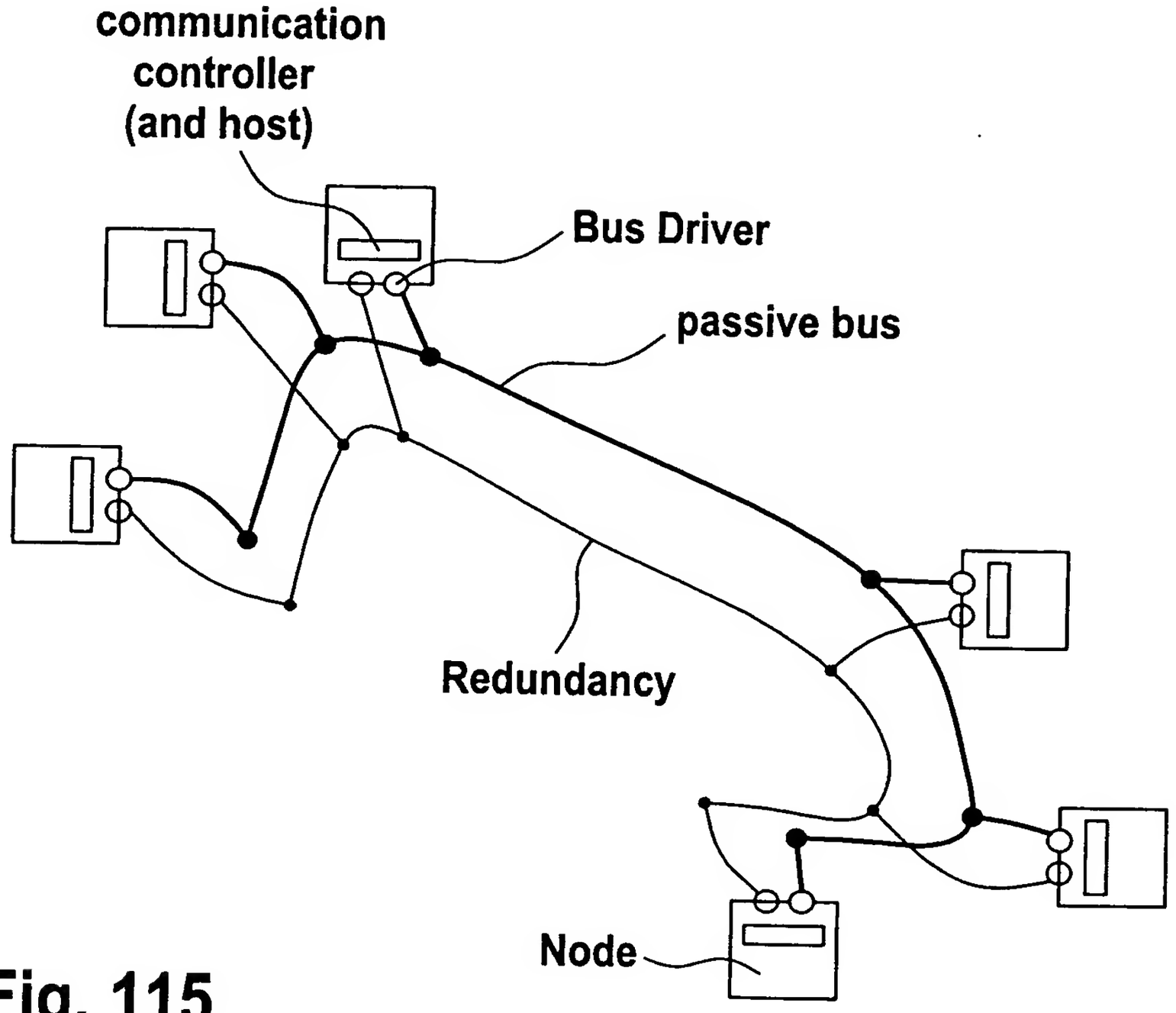


Fig. 115

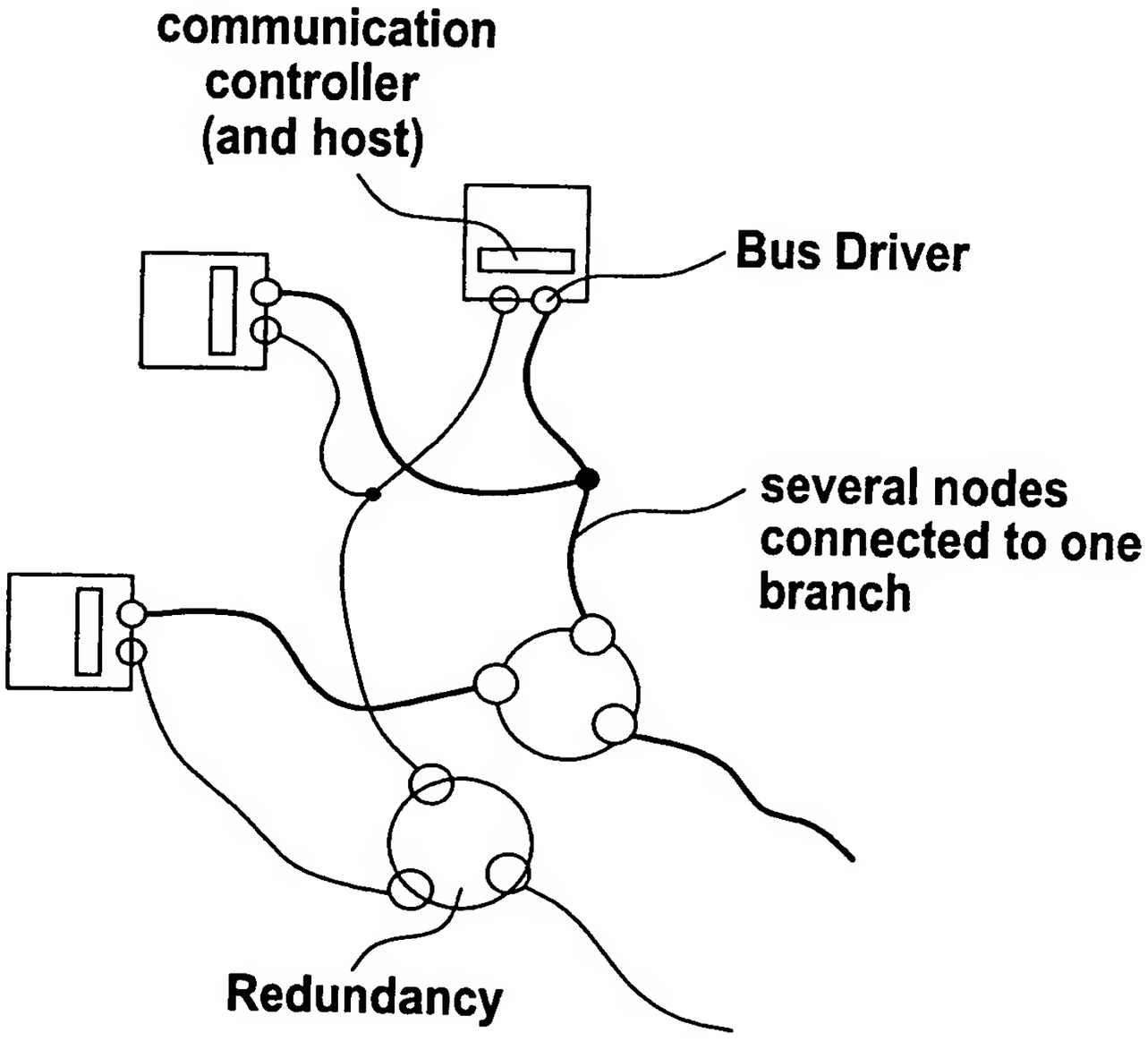


Fig. 116

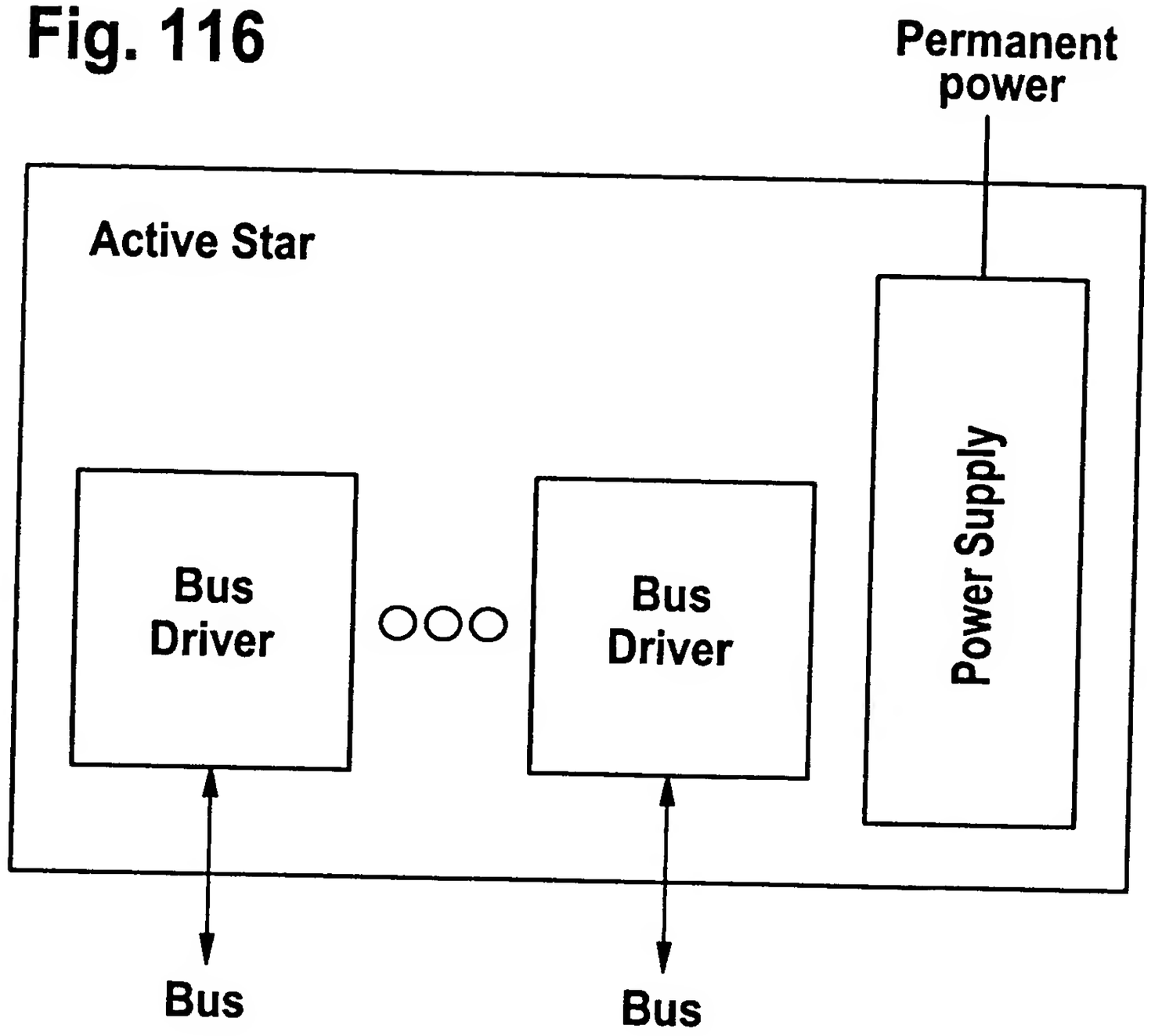
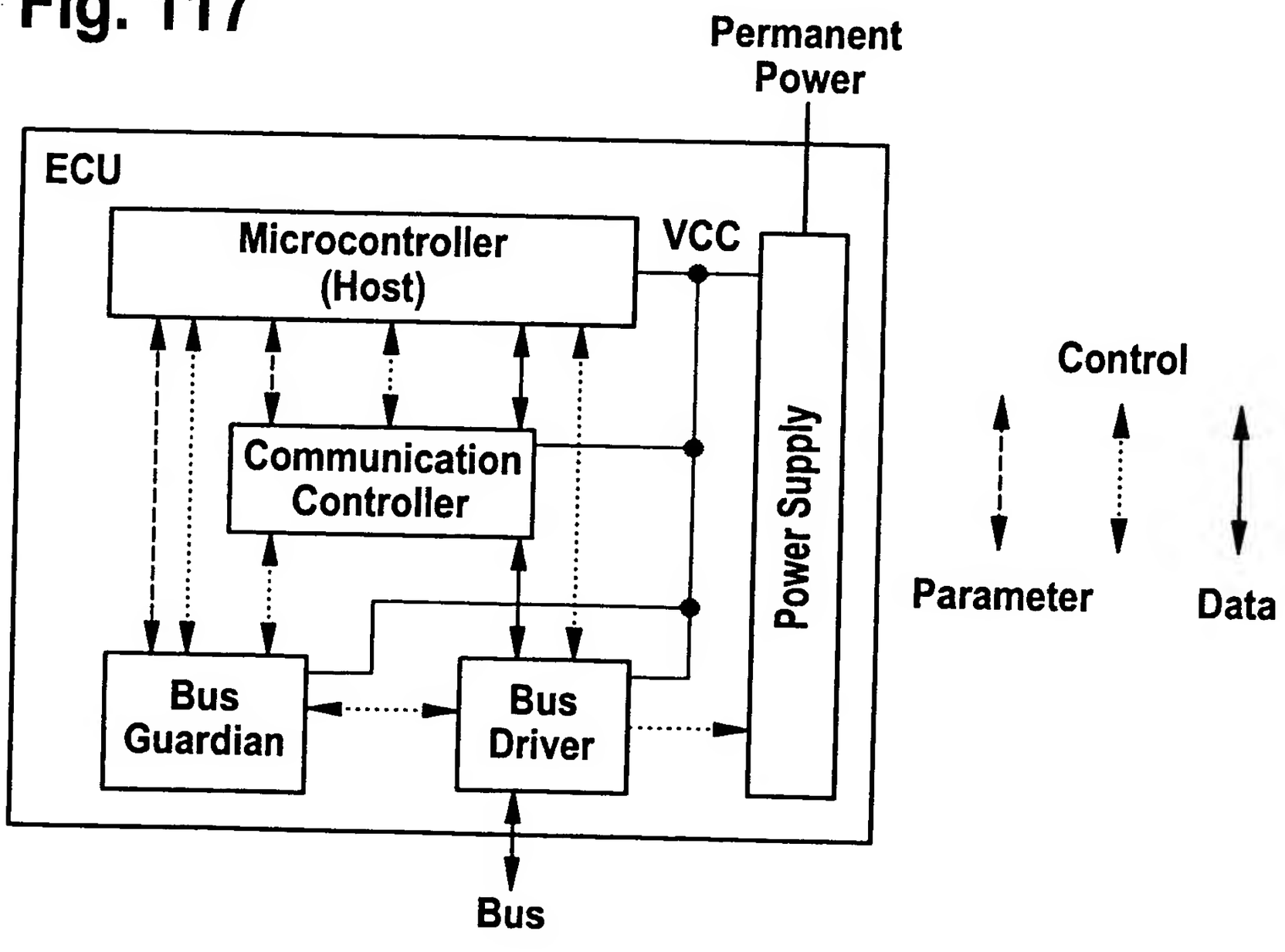



Fig. 117